



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

TEMA:

“SISTEMA IOT DE MONITOREO DE CALIDAD DEL AIRE DE CONCENTRACIONES
DE NO₂ Y O₃ CON REDES NEURONALES EN LA CIUDAD DE IBARRA”

**Trabajo de titulación previo a la obtención del título de Ingeniero en
Telecomunicaciones**

Línea de investigación: Producción industrial y tecnología sostenible

AUTOR:

Michael Israel Negrete Ramírez

DIRECTOR:

MSc. Fabián Geovanny Cuzme Rodríguez

Ibarra – Ecuador 2026



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1004842579		
APELLIDOS Y NOMBRES:	Negrete Ramírez Michael Israel		
DIRECCIÓN:	Rio Quinindé y Hernán Gonzales de Saa		
EMAIL:	minegreter@utn.edu.ec / ir460787@gmail.com		
TELÉFONO FIJO:		TELÉFONO MÓVIL:	0962077234

DATOS DE LA OBRA	
TÍTULO:	SISTEMA IOT DE MONITOREO DE CALIDAD DEL AIRE DE CONCENTRACIONES DE NO2 Y O3 CON REDES NEURONALES EN LA CIUDAD DE IBARRA
AUTOR (ES):	NEGRETE RAMIREZ MICHAEL ISRAEL
FECHA: DD/MM/AAAA	21/01/2026
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN TELECOMUNICACIONES
DIRECTOR:	MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ
ASESOR:	MSC. JAIME ROBERTO MICHILENA CALDERÓN

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrollo, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por partes de terceros.

Ibarra, a los 21 días del mes de enero de 2026.

EL AUTOR:

Michael Israel Negrete Ramírez

**CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE
INTEGRACIÓN CURRICULAR**

Ibarra, 21 de enero del 2026

MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

(f)

MSc. Fabián Geovanny Cuzme Rodríguez

C.C.: 1311527012

DEDICATORIA

Dedico este trabajo, en primer lugar, a Dios, fuente de luz en los días nublados y guía silenciosa en los momentos de incertidumbre, presencia constante en mis desvelos y fortaleza en cada paso dado.

A mi madre, Nelva Ramírez, raíz firme de mi vida y manantial inagotable de amor. Su voz ha sido abrigo, y su fe, el impulso que me sostuvo cuando el cansancio quiso vencerme.

A mi padre, Francisco Negrete, quien me enseñó a caminar con la verdad, a sostener la palabra y actuar con rectitud, con la responsabilidad y coherencia que dan sentido a cada acción.

A mis hermanos, por acompañarme en cada paso y brindarme su fortaleza de manera discreta pero inquebrantable.

A mi amiga Samantha Pérez, por su apoyo constante, su palabra honesta y su compañía a lo largo de este camino.

Este logro ha sido posible gracias al respaldo de mi familia, la orientación de mis docentes y el apoyo de amigos que contribuyeron a lo largo de este proceso.

“Ningún esfuerzo es individual; todo es colectivo.”

Negrete Ramírez Michael Israel

AGRADECIMIENTO

Agradezco primeramente a Dios, por ser mi guía constante, por fortalecerme en los momentos de incertidumbre y acompañarme con fe en cada paso de este recorrido académico.

A mi tutor de tesis, MSc. Fabián Cuzme, por su valiosa orientación, paciencia y compromiso durante el desarrollo de este trabajo. Su experiencia y apoyo fueron fundamentales para lograr este objetivo.

Al MSc. Jaime Michilena, asesor del trabajo, por sus aportes técnicos y sugerencias acertadas que enriquecieron el contenido de este estudio.

A todos quienes, de una u otra manera, formaron parte de este proceso, mi más sincero agradecimiento.

Negrete Ramírez Michael Israel

RESUMEN

El presente trabajo expone el desarrollo de un sistema IoT de monitoreo de calidad del aire enfocado en la medición de dióxido de nitrógeno (NO_2) y ozono (O_3) en la ciudad de Ibarra, integrando tecnologías de Internet de las Cosas (IoT) e inteligencia artificial mediante redes neuronales artificiales. La investigación se ejecutó bajo la metodología Bottom-Up y el estándar IEEE 29148, estableciendo los requerimientos de hardware y software necesarios para asegurar la correcta integración de los módulos de adquisición, procesamiento y comunicación del sistema. El prototipo fue implementado en una arquitectura modular que permite la recolección y transmisión inalámbrica de datos hacia una plataforma web en tiempo real. Durante las pruebas de campo, el sistema demostró un funcionamiento estable y confiable, alcanzando una autonomía energética de 18 horas y una transmisión de datos continua sin pérdidas.

En el procesamiento de la información ambiental se empleó el lenguaje de programación Python junto con bibliotecas especializadas de aprendizaje automático. Se diseñó e implementó una red neuronal multicapa, entrenada durante 100 épocas, que alcanzó una precisión del 97 % en la clasificación de los niveles de calidad del aire. Los resultados experimentales mostraron concentraciones promedio entre 0.00 y 0.03 ppm de NO_2 ($\approx 0\text{--}57 \mu\text{g}/\text{m}^3$) y 15 a 33 ppb de O_3 ($\approx 30\text{--}65 \mu\text{g}/\text{m}^3$), valores inferiores a los límites establecidos por la Organización Mundial de la Salud (OMS), clasificando la calidad del aire como “Buena” en todos los puntos de monitoreo.

Palabras clave: Internet de las Cosas, red neuronal artificial, calidad del aire, dióxido de nitrógeno, ozono, monitoreo ambiental, inteligencia artificial.

ABSTRACT

This paper presents the development of an IoT air quality monitoring system focused on measuring nitrogen dioxide (NO₂) and ozone (O₃) in the city of Ibarra, integrating Internet of Things (IoT) technologies and artificial intelligence through artificial neural networks. The research was carried out using the Bottom-Up methodology and the IEEE 29148 standard, establishing the hardware and software requirements necessary to ensure the correct integration of the system's acquisition, processing, and communication modules. The prototype was implemented in a modular architecture that allows for the collection and wireless transmission of data to a web platform in real time. During field tests, the system demonstrated stable and reliable operation, achieving 18 hours of energy autonomy and continuous data transmission without loss.

The Python programming language was used to process environmental information, together with specialized machine learning libraries. A multilayer neural network was designed and implemented, trained for 71 epochs, which achieved 97% accuracy in classifying air quality levels. The experimental results showed average concentrations between 0.00 and 0.03 ppm of NO₂ ($\approx 0\text{--}57 \mu\text{g}/\text{m}^3$) and 15 to 33 ppb of O₃ ($\approx 30\text{--}65 \mu\text{g}/\text{m}^3$), values below the limits established by the World Health Organization (WHO), classifying air quality as “Good” at all monitoring points.

Keywords: Internet of Things, Artificial Neural Network, air quality, nitrogen dioxide, ozone, environmental monitoring, artificial intelligence.

LISTA DE SIGLAS

- AI.** Artificial Intelligence (Inteligencia Artificial)
- AQI.** Air Quality Index (Índice de Calidad del Aire)
- EPA.** Environmental Protection Agency (Agencia de Protección Ambiental de EE. UU.)
- GAD.** Gobierno Autónomo Descentralizado
- Gb.** Gigabyte
- GPS.** Global Positioning System (Sistema de Posicionamiento Global)
- GPRS.** General Packet Radio Service (Servicio General de Paquetes vía Radio)
- HTTP.** Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
- ICA.** Índice de Calidad del Aire
- IDE.** Integrated Development Environment (Entorno de Desarrollo Integrado)
- IEEE.** Institute of Electrical and Electronics Engineers
- IoT.** Internet of Things (Internet de las Cosas)
- MLP.** Multi-Layer Perceptron (Perceptrón Multicapa)
- NO₂.** Dióxido de Nitrógeno
- O₃.** Ozono
- OMS.** Organización Mundial de la Salud
- PPB.** Partes por Billón
- PPM.** Partes por Millón
- RAM.** Random Access Memory (Memoria de Acceso Aleatorio)
- ReLU.** Rectified Linear Unit (Unidad Lineal Rectificada)
- UART.** Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal)
- USB.** Universal Serial Bus
- Wi-Fi.** Wireless Fidelity (Conectividad Inalámbrica)

ÍNDICE DE CONTENIDOS

CAPÍTULO I: ANTECEDENTES	1
1.1. Tema.....	1
1.2. Problema	1
1.3. Objetivos	4
1.3.1. Objetivo General.....	4
1.3.2. Objetivos Específicos.....	4
1.4. Alcance.....	5
1.5. Justificación	7
CAPÍTULO II: MARCO TEÓRICO.....	10
2.1. Contaminación del Aire	10
2.1.1 Material Particulado PM10 y PM2.5	11
2.1.2 Variables de contaminantes atmosféricos.	12
2.2. Internet de las cosas (IoT).	16
2.2.1. Arquitectura de Sistemas IoT.....	17
2.2.2 Tecnologías de conectividad en IoT	19
2.2.3. Rendimiento de los sistemas IoT en el Monitoreo de la calidad del aire. .	20
2.3. Redes Neuronales Artificiales (ANN).	20
2.3.1. Estructura y funcionamiento de las Redes Neuronales.....	21
2.3.2. Tipos de redes Neuronales.	22
2.3.3. Proceso de Desarrollo de ANN.....	25

2.3.4. Tipos de redes neuronales más usadas en sistemas de calidad del aire.	30
CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.....	35
3.1. Metodología.....	35
3.2. Análisis de la situación actual.	36
3.3. Requerimientos del sistema.	36
3.3.1. Nomenclatura de requerimientos	36
3.3.2. Identificación de Stakeholders.....	37
3.3.3. Requerimientos de Stakeholders.....	37
3.3.4. Requerimientos del sistema.	39
3.3.5. Requerimientos de arquitectura	41
3.4. Selección de Hardware y Software del sistema.	43
3.4.1. Selección de Hardware.....	43
3.4.2. Selección de Software.....	52
3.5. Formación de dataset.	53
3.5.1. Categorización de variables ambientales según niveles de riesgo.	54
3.6. Diseño del sistema	55
3.6.1. Arquitectura IoT del Sistema	55
3.6.2. Diagrama de Bloques General del Sistema.....	56
3.6.3. Diagramas de Flujo.	58
3.6.4. Comunicación del sistema.	64
3.6.5. Diseño general de interfaz Web.	65

3.6.6. Plataforma de despliegue de interfaz web.....	67
3.6.7. Diagrama esquemático de conexiones del sistema.	67
3.6.8. Consumo energético del sistema.....	69
3.6.9. Diseño del Case del sistema.	70
3.7. Implementación del sistema.....	71
3.7.1. Construcción de modelo de red neuronal.....	72
3.7.2. Implementación del software.....	82
3.7.3 Implementación del hardware.....	94
CAPÍTULO IV: PRUEBAS Y RESULTADOS.....	99
4.1 Pruebas del sistema.	99
4.1.2 Pruebas Individuales de componentes del sistema.	102
4.1.3 Pruebas del sistema completo	109
4.1.4 Resultados de pruebas individuales y sistema completo	115
4.3 Pruebas de integración del sistema completo en funcionamiento	118
4.3.1 Designación de puntos de evaluación del sistema.	119
4.3.2 Prueba de campo del sistema.	120
4.4 Resultados.....	121
4.4.1 Resultados obtenidos en la ciudad de Ibarra.....	121
4.4.2 Resultados obtenidos en la ciudad de Quito	125
4.6 Comparación y análisis de resultados.....	128
4.6.1 Ibarra	128

4.6.2 Quito.....	131
4.7 Análisis Costo/Beneficio	134
CONCLUSIONES Y RECOMENDACIONES	143
CONCLUSIONES.....	143
RECOMENDACIONES.....	145
Bibliografía	146
ANEXOS	159

ÍNDICE DE TABLAS

Tabla 1 Aplicación de ANN en el Monitoreo y Predicción de la Calidad del Aire	31
Tabla 2 Nomenclatura de Requerimientos.....	37
Tabla 3 Lista de participantes en el proyecto.....	37
Tabla 4 Requerimientos de Stakeholders.....	38
Tabla 5 Requerimientos del sistema.	39
Tabla 6 Requerimientos de Arquitectura	41
Tabla 7 Selección del microcontrolador	43
Tabla 8 Selección de la placa de procesamiento.....	44
Tabla 9 Selección de sensor NO2	46
Tabla 10 Selección de sensor O3	48
Tabla 11 Selección de sensor de temperatura y humedad.	49
Tabla 12 Selección de sensor de GPS.....	51
Tabla 13 Selección de framework de desarrollo de interfaz web.	52
Tabla 14 Selección de lenguaje de programación.....	53
Tabla 15 Rangos de niveles de contaminación.	55

Tabla 16 Consumo energético de los componentes.	69
Tabla 17 Desempeño global de predicción	80
Tabla 18 Nomenclatura de pruebas.....	99
Tabla 19 Pruebas realizadas para la validación del sistema.....	100
Tabla 20 Resultados de pruebas individuales de los componentes.	115
Tabla 21 Puntos de monitoreo	119
Tabla 22 Tabla de resultados Ibarra - 14 al 18 de Julio 2025	122
Tabla 23 Tabla de resultados Ibarra - 21 al 25 de Julio 2025	123
Tabla 24 Tabla comparativa de resultados de sistema diseñado y estación oficial ..	126
Tabla 25 Costos de Hardware para el sistema.	135
Tabla 26 Costos de software del sistema.	137
Tabla 27 Costos de ingeniería para el sistema.	139
Tabla 28 Costos totales para el sistema	141

ÍNDICE DE FIGURAS

Figura 1 Árbol de problemas	3
Figura 2 Metodología Planeada para la realización del proyecto	5
Figura 3 Arquitectura genérica del sistema	7
Figura 4 Arquitectura IoT	17
Figura 5 Estructura Básica de una red neuronal	21
Figura 6 Diagrama de procedimiento de ANN.....	25
Figura 7 Arduino MEGA Mini 250 Pro.....	44
Figura 8 Raspberry Pi Zero 2W	45

Figura 9 Gravity: Factory Calibrated Electrochemical Nitrogen Dioxide Sensor	47
Figura 10 Gravity: Electrochemical Ozone Sensor	48
Figura 11 Gravity: I2C BME280 Environmental Sensor	50
Figura 12 GPS + BDS BeiDou Dual Module.....	51
Figura 13 Arquitectura IoT del Sistema.	56
Figura 14 Diagrama de Bloques General del Sistema.	57
Figura 15 Diagrama de Flujo de procesamiento de datos.....	58
Figura 16 Diagrama de Flujo de entrenamiento de la red Neuronal.....	60
Figura 17 Diagrama de Flujo del sistema.	62
Figura 18 Diagrama de secuencia del sistema.	64
Figura 19 Diseño general de pantalla predeterminada.....	65
Figura 20 Diseño general de pantalla de resultados.	66
Figura 21 Diagrama esquemático de conexiones del sistema.	68
Figura 22 Diseño de case del sistema.	71
Figura 23 Visualización de Separabilidad de Clases.	73
Figura 24. Escalado de datos antes de la división.....	74
Figura 25 División de datos de entrenamiento.	75
Figura 26 Arquitectura general del modelo.	76
Figura 27 Parámetros de compilación del modelo.....	76
Figura 28 Funciones de control del entrenamiento.	77
Figura 29 Entrenamiento del modelo.....	77
Figura 30 Gráficos del comportamiento del modelo en 100 épocas.....	78
Figura 31 Gráficos del comportamiento del modelo en 40 épocas.	78
Figura 32 Gráficos del comportamiento del modelo en 20 épocas.....	79
Figura 33 Conversión del modelo a tflite.	81

Figura 34 Exportación del escalador utilizado.....	82
Figura 35 Inicialización de sensores y calentamiento.....	83
Figura 36 Función de recolección de datos y archivo JSON.....	85
Figura 37 Importación de librerías.....	86
Figura 38 Credenciales y carga de datos.....	86
Figura 39 Procesamiento de datos.	87
Figura 40 Envío de datos a Servidor.....	87
Figura 41 Recolección de datos del servidor.	88
Figura 42 Estructura de datos en firestore.	90
Figura 43 Organización del proyecto.....	91
Figura 44 Pantalla principal de interfaz web.	92
Figura 45 Pantalla de mediciones de interfaz web.....	93
Figura 46 Pantalla de mediciones resumen.	93
Figura 47 Batería externa WGP de 10 400 mAh.	94
Figura 48 Primera capa interna del dispositivo.	95
Figura 49 Segunda capa interna del dispositivo.	96
Figura 50 Parte frontal del dispositivo.....	97
Figura 51 Sensores encendidos.....	103
Figura 52 Prueba de sensores en campo abierto dentro del case.	104
Figura 53 Prueba de sensores en fuente directa de NOx dentro del case.	105
Figura 54 Recepción de datos de Arduino a Raspberry Pi Zero 2W.	106
Figura 55 Datos procesados por Raspberry Pi Zero 2W.	106
Figura 56 Validación de datos y Firestone en Raspberry Pi Zero 2W.	107
Figura 57 Prueba del case de protección bajo condiciones húmeda.	108
Figura 58 Plan gratuito Netlify de página web.....	108

Figura 59	Envío de datos erróneos de Arduino Mega a Raspberry Pi Zero 2W.	109
Figura 60	Conexión móvil interrumpida en Raspberry Pi Zero 2W.....	110
Figura 61	Prueba de velocidad de procesamiento en Raspberry Pi Zero 2W.....	111
Figura 62	Interfaz web con dimensiones de smarthphone.....	114
Figura 63	Puntos de monitoreo de calidad del aire en la ciudad de Ibarra.	120
Figura 64	Desarrollo de las pruebas en las 5 zonas designadas.....	121
Figura 65	Gráfico de Resultados de Dióxido de nitrógeno Ibarra.....	124
Figura 66	Gráfico de Resultados de Ozono Ibarra.	124
Figura 67	Gráfico de Resultados de temperatura Ibarra.	124
Figura 68	Gráfico de resultados de porcentajes de humedad Ibarra.	124
Figura 69	Monitoreo realizado en centro histórico de Quito.....	125
Figura 70	Variación horaria de NO ₂ en diferentes sectores de Quito.....	127
Figura 71	Variación horaria de O ₃ en diferentes sectores de Quito.....	127
Figura 72	Resultados de monitoreo de Ozono año 2015.....	129
Figura 73	Resultados de monitoreo de Dióxido de nitrógeno año 2015.....	129
Figura 74	Gráfico comparativo NO ₂ Sistema Diseñado y monitoreo 2015.....	131
Figura 75	Gráfico comparativo O ₃ Sistema Diseñado y monitoreo 2015.....	131
Figura 76	Gráfico comparativo NO ₂ Sistema Diseñado y Estación oficial Quito ...	132
Figura 77	Gráfico comparativo O ₃ Sistema Diseñado y Estación oficial Quito	132

CAPÍTULO I: ANTECEDENTES

En este capítulo se presenta el contexto y la problemática relacionada con los sistemas de monitoreo de calidad del aire en la ciudad de Ibarra, con un enfoque en la medición de los contaminantes NO₂ y O₃. Se establecen los objetivos, tanto generales como específicos, que definen el propósito y las metas del proyecto, delimitando el alcance y fundamentando su justificación.

1.1. Tema

SISTEMA IOT DE MONITOREO DE CALIDAD DEL AIRE DE CONCENTRACIONES DE NO₂ Y O₃ CON REDES NEURONALES EN LA CIUDAD DE IBARRA

1.2. Problema

Las estimaciones realizadas por la Organización Mundial de la Salud (OMS) en el 2019 informo que alrededor de 4.2 millones de muertes en las ciudades y zonas rurales son debido a contaminantes del aire de exteriores (World Health Organization, 2022). Y se corrobora en las directrices mundiales de la OMS sobre la calidad del aire que las enfermedades causadas por exposiciones a la contaminación del aire son reconocidas como la amenaza medioambiental más peligrosa para la salud humana (Organización Mundial de la Salud, 2021).

En el Plan Nacional de la Calidad 2023 se menciona que “En el país hay poca investigación sobre los efectos de la contaminación del aire en la salud de las personas, estos temas no se han incluido en los programas de desarrollo urbano y no se han realizado estudios epidemiológicos relacionados con la contaminación del aire. Las principales causas de la contaminación del aire son los procesos industriales y de automoción” (Ministerio de Producción, Comercio Exterior, Inversiones y Pesca, 2023, p. 65).

Y en el entorno local de la ciudad de Ibarra el último monitoreo de la calidad del aire fue durante los años 2012 – 2015 en la cual se la clasificaba como la primera ciudad del Ecuador con el Aire más limpio, sin embargo, el ministerio del Ambiente no proporciono datos que certifiquen dicha clasificación del estado actual de la calidad del aire. Debido al deterioro de los equipos no se ha continuado realizado otro monitoreo hasta la fecha (GAD Municipal de San Miguel de Ibarra, 2022).

Según las problemáticas prospectivas del Plan de Desarrollo, el escaso monitoreo de la calidad del aire podría resultar en una disminución significativa de su calidad debido al crecimiento poblacional y vehicular. De acuerdo con el VII Censo Nacional de Población llevado a cabo en el año 2010, la población total del cantón Ibarra alcanzaba los 181.175 habitantes, con una proporción de 128 vehículos matriculados por cada 1000 habitantes (INEC, 2010). Y según proyecciones se tiene un crecimiento del 1.60% anualmente en la ciudad de Ibarra (GAD Municipal de San Miguel de Ibarra, 2022).

Debido a la capacidad de ciertos contaminantes atmosféricos para interactuar y propagarse rápidamente, es crucial centrarse en la detección prioritaria de los gases altamente reactivos, como el ozono (O₃) y los compuestos de nitrógeno (NO_x, NO_y), clasificados como PM 2.5. Según Telégrafo (2014), la ciudad de Ibarra registró una concentración de 9 microgramos de PM 2.5 por metro cúbico anualmente durante el 2014, mientras que el límite establecido en la Norma de Calidad del Aire Ambiental era de 15 microgramos para no ser nocivo para la salud.

Cabe indicar que para una contaminación atmosférica significativa y potencialmente peligrosa son los gases de Dióxido de nitrógeno (NO₂) y Ozono (O₃) ya que pueden generar diversos impactos en el sistema respiratorio. Dado que el parque automotor es la principal

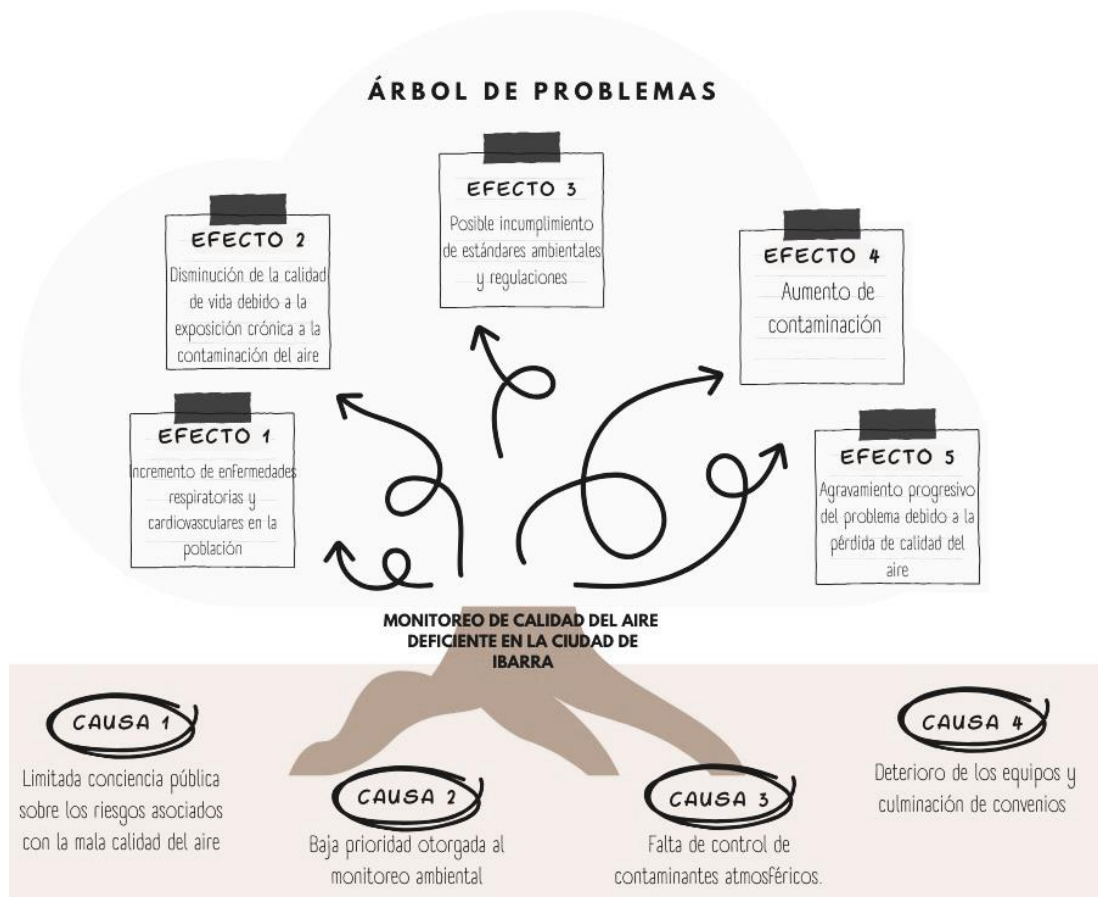
fuente externa de dióxido de nitrógeno, las zonas altamente pobladas son particularmente vulnerables a sus impactos negativos (American Lung Association, 2023). Y el ozono troposférico, también conocido como O₃, se forma cuando los óxidos de nitrógeno (NO_x) y los compuestos orgánicos volátiles (COV) interactúan químicamente bajo la influencia de la radiación solar siendo también emitidos estos contaminantes por vehículos, centrales eléctricas, etc. (United States Environmental Protection Agency, 2024).

La falta de equipos de medición por parte de las autoridades y apoyo del Ministerio del ambiente ha generado una carencia en el conocimiento sobre la calidad del aire actual en la ciudad. La adquisición de equipos de monitoreo está restringida por el presupuesto y falta de proyectos por los costos que genera su administración. Existen un reducido número de dispositivos que utilicen una recopilación de datos de la calidad del aire y detecten niveles de la contaminación con ayuda de la implantación de una red neuronal para lecturas de sensores más detalladas (Bombela Jiménez et al., 2019).

El propósito es desarrollar un sistema IoT de monitoreo de calidad del aire enfocado a la medición de NO₂ y O₃ usando una red neuronal que mejora la recopilación de datos de los sensores para aplicarlo en los puntos de mayor concurrencia vehicular y poblacional de la ciudad de Ibarra, y así determinar si el estado actual del aire es o no perjudicial para la salud.

Figura 1

Árbol de problemas



Nota. Autoría Propia

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar un sistema IoT de monitoreo de calidad del aire que permita medir niveles de concentración de NO₂ y O₃ en puntos de mayor aglomeración de vehículos y personas en la ciudad de Ibarra mediante el uso de red neurona.

1.3.2. Objetivos Específicos

- Recolectar información bibliográfica sobre los contaminantes PM 2.5 orientado a el Dióxido de Nitrógeno y Ozono, y Redes neuronales.

- Determinar los requerimientos de software y hardware del sistema, utilizando las buenas prácticas y recomendaciones establecidas por estándares o normativas.
- Diseñar el sistema, usando sensores de NO₂ y O₃ que integre la red neuronal mediante una interfaz web que permita visualizar el nivel de concentración en el aire.
- Realizar pruebas de funcionamiento del sistema en puntos designados de la ciudad de Ibarra para verificación del rendimiento.

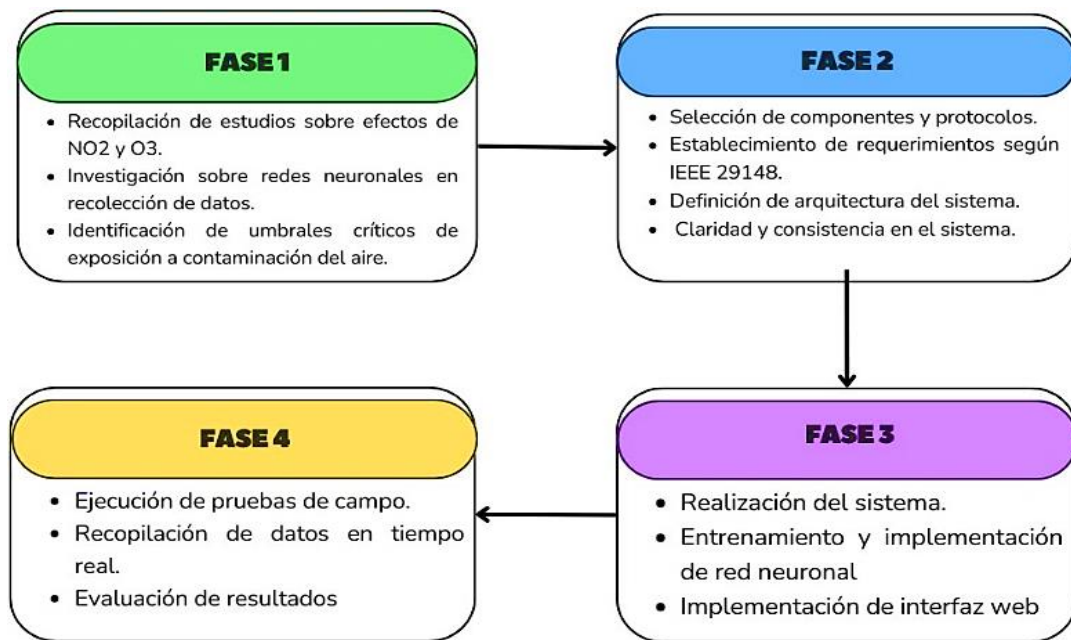
1.4. Alcance

El presente proyecto tiene como objetivo principal desarrollar un sistema IoT de monitoreo de calidad del aire orientado a la medición de NO₂ y O₃ con red neuronal para puntos de mayor concurrencia vehicular y de personas en la Ciudad de Ibarra, con el fin de conocer el estado actual de la contaminación del aire. El proyecto se realizará con la metodología Bottom Up, en la que se parte de algo pequeño que crece hasta llegar a un sistema terminado y complejo (Restrepo Muñóz, 2009).

Las cuatro fases de la metodología bottom up, se representan de forma resumida en la figura 2 y se detalla cada una a continuación:

Figura 2

Metodología Planeada para la realización del proyecto



Nota. Autoría Propia

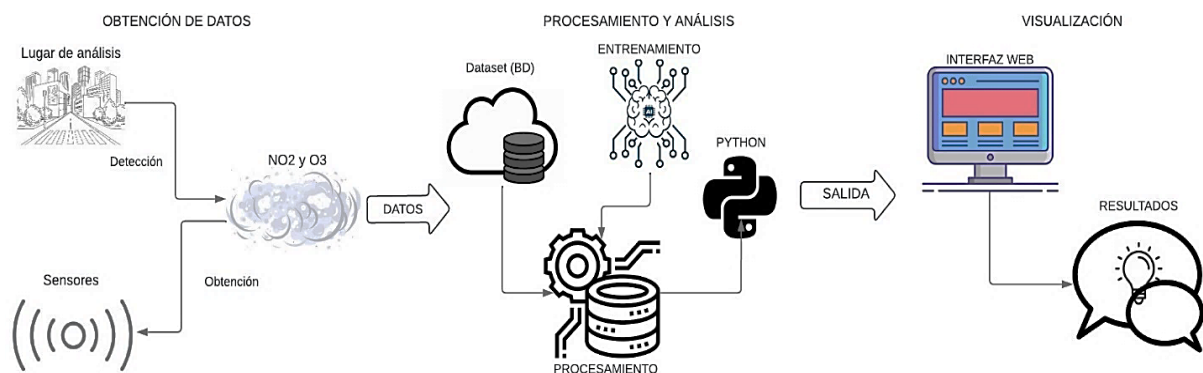
En la fase de especificaciones, se llevará a cabo una investigación para la obtención de información de estudios relevantes sobre los efectos adversos del dióxido de nitrógeno (NO₂) y ozono (O₃) en la calidad del aire, y redes neuronales aplicadas a recopilación de datos de sensores. El objetivo de esta investigación es conocer los umbrales de exposición considerados peligrosos para la salud humana y el medio ambiente, siguiendo las directrices de organismos reguladores y estudios científicos reconocidos.

En la fase de ciclo de verificación y simulación, se procederá a definir la arquitectura del sistema seleccionando los componentes adecuados y protocolos de comunicación necesarios para la integración del sistema. Además, se establecerá los requerimientos detallados del software *open source*, siguiendo el estándar IEEE 29148 para garantizar su claridad y consistencia.

En la Figura 3 que se muestra a continuación, se representa en tres etapas generales un modelo preliminar de arquitectura para el diseño del sistema:

Figura 3

Arquitectura genérica del sistema



Nota. Autoría Propia

En la fase de integración de bloques, se procederá a implementar el sistema IoT de monitoreo de calidad del aire, mediante el desarrollo y entrenamiento de las redes neuronales. Además, implementar una interfaz web para visualizar y presentar los datos recopilados de manera intuitiva y accesible.

En la fase de verificación del sistema, se seleccionarán puntos en la Ciudad de Ibarra que representen adecuadamente las condiciones de concurrencia vehicular y de personas. Estos puntos se utilizarán para realizar pruebas de campo para validar el sistema de monitoreo de calidad del aire. Durante estas pruebas, se recopilarán datos en tiempo real sobre las concentraciones de NO2 y O3 en el aire, lo que permitirá obtener resultados que reflejen de manera cercana a la realidad las concentraciones de contaminantes en el aire de la Ciudad de Ibarra.

1.5. Justificación

En el marco de los Objetivos de Desarrollo Sostenible (ODS) de la ONU, se plantea en el Objetivo 11 sobre ciudades y comunidades sostenibles, específicamente en la Meta 11.6, la necesidad de reducir, para el año 2030, el impacto ambiental negativo per cápita de las áreas urbanas. Esto implica una atención particular hacia la calidad del aire y la gestión adecuada de

los desechos municipales y otros residuos (Organización de las Naciones Unidas, 2023). En el entorno Nacional se tiene la norma de calidad del Aire del ambiente o Nivel de Inmisión adjunta en el Texto Unificado de la Legislación Secundaria del Ministerio del Ambiente (TULSMA). La normativa establece parámetros para las concentraciones de contaminantes tanto criterio en el aire ambiente como para aquellos contaminantes no convencionales que poseen efectos tóxicos y/o cancerígenos en el mismo. (TULSMA, 2015).

Y en el Plan Nacional de la Calidad (2023) se menciona que se requiere la continuación de un nuevo convenio como el que se tenía en años atrás para poder optimizar la medición de material particulado PM 2,5 y PM10 (pág.65). Y en la Actualización del plan de desarrollo y ordenamiento territorial del cantón Ibarra 2015-2023 se destaca la necesidad imperativa de actualizar la Ordenanza para la Protección de la Calidad Ambiental. Esta actualización es fundamental para abordar la contaminación derivada de desechos no domésticos generados por fuentes fijas. Además, se plantea la expansión de esta ordenanza para incluir su aplicación a fuentes móviles. El propósito es implementar medidas preventivas que garanticen la óptima calidad del aire, promoviendo el bienestar de la población (GAD Municipal de San Miguel de Ibarra., 2015).

En el análisis de los datos de monitoreo de calidad del aire realizado en la ciudad de Ibarra entre 2012 - 2015, Loachamín Tipán (2017) describió:

El crecimiento poblacional y el incremento del parque automotor pueden generar progresivamente niveles críticos de concentración de contaminantes atmosféricos que pueden ocasionar efectos nocivos para la salud de sus habitantes (pág. 46).

Por lo tanto, ya que antes Ibarra no presentaba niveles significativos de concentraciones de material particulado 2,5 en 2015, por el crecimiento de población experimentado hasta el presente año, es crucial tener información actualizada sobre la contaminación del aire en la ciudad. Sobre todo, a que no se ha realizado nuevos monitores como se menciona en el Plan de

Desarrollo y Ordenamiento Territorial (2022) debido a problemas con equipos y la finalización de convenios con el Ministerio del Ambiente, no se cuentan con nuevos datos al respecto (pág. 41).

De esta manera, el proyecto a desarrollar será un valioso aporte a la comunidad, ya que permitirá monitorear el estado de la contaminación del aire, centrándose en los niveles de NO₂ y O₃, los cuales la Organización Mundial de la Salud (OMS) considera indicadores claros de la contaminación por partículas PM 2.5 (World Health Organization, 2022). Por otra parte, como mencionan De Jesús et al. (2016) en los sistemas de redes neuronales para sistemas de monitoreo ambiental resulta aceptable incorporar algoritmos de redes neuronales en sistemas portables.

Además, al emplear redes neuronales, se beneficia de que la mayor parte del procesamiento intensivo ocurre durante el entrenamiento inicial. Una vez que la red neuronal ha sido entrenada para realizar una tarea específica, su ejecución es rápida, lo que permite identificar muestras desconocidas de manera eficiente (Ludermir & Yamazaki, 2003). Por ello, se implementarán redes neuronales al sistema de monitoreo de calidad del aire, lo que garantiza una eficaz detección y análisis de la contaminación del aire en tiempo real y ofrece una solución práctica y versátil para su implementación en entornos donde se requiera el análisis de variables ambientales

CAPÍTULO II: MARCO TEÓRICO

En este capítulo se presenta el marco teórico que fundamenta el presente trabajo, proporcionando la base necesaria para el análisis y la comprensión del proyecto. Se analizará la contaminación del aire centrada en el material particulado (PM2.5, PM10) y gases como NO₂ y O₃, evaluando sus orígenes y los límites de salud establecidos por la OMS. También se abordarán los sistemas IoT y las redes neuronales aplicadas a la monitorización y predicción ambiental.

Se proporcionará una visión general del proceso de construcción y entrenamiento de estas redes, y los tipos de redes neuronales aplicadas en este contexto, proporcionando una visión general del proceso de construcción y entrenamiento de estas redes. Se destacará su importancia y efectividad en la mejora de los sistemas de gestión de la calidad del aire.

2.1. Contaminación del Aire

La contaminación atmosférica se define como la presencia de sustancias nocivas en la atmósfera en concentraciones que pueden dañar la salud de la población o afectar diversos ecosistemas (Álvarez, 2011; Ubilla & Yohannessen, 2017). El desarrollo económico y demográfico de la sociedad depende en gran medida de la explotación de los recursos naturales; sin embargo, esta práctica ha degradado el medio ambiente y ha contribuido significativamente a la contaminación del aire (Santurtún et al., 2017).

Los efectos adversos del cambio climático han destacado la estrecha relación entre la protección del medio ambiente y la salud humana, reconocida por la Constitución del Ecuador. Pese a ser un requisito básico según la OMS, la mala calidad del aire persiste como una de las mayores amenazas para la salud pública mundial (World Health Organization, 2021).

Las principales fuentes de contaminación en Ecuador abarcan la mala planificación urbana e industrial, tecnologías anticuadas, combustibles deficientes y actividades extractivas como la minería (Celi Ortega, 2018).

2.1.1 Material Particulado PM10 y PM2.5

El material particulado (PM, por sus siglas en inglés) se define como una combinación heterogénea de partículas sólidas y gotas líquidas que permanecen en suspensión en la atmósfera. Estas partículas varían en tamaño, composición y origen, lo que influye en el tiempo que permanecen en suspensión (Quijano Parra et al., 2010). Principalmente, el material particulado se clasifica en dos categorías: PM10 y PM2.5.

PM10 se refiere a partículas con un diámetro aerodinámico de 10 micrómetros o menos. Estas partículas son lo suficientemente pequeñas como para ser inhaladas y pueden penetrar en las vías respiratorias superiores, afectando la salud respiratoria y cardiovascular. Las fuentes comunes de PM10 incluyen el polvo de carreteras, la construcción, la agricultura, y las emisiones industriales (Quijano Parra et al., 2010).

PM2.5, por otro lado, se refiere a partículas con un diámetro aerodinámico de 2.5 micrómetros o menos. Debido a su tamaño más pequeño, PM2.5 puede penetrar más profundamente en los pulmones y entrar en el torrente sanguíneo, lo que puede provocar problemas de salud más graves, como enfermedades cardiovasculares, enfermedades respiratorias y cáncer de pulmón. Las principales fuentes de PM2.5 incluyen la quema de combustibles fósiles, las emisiones de vehículos, la quema de biomasa y ciertas actividades industriales (Lyons et al., 2024).

Según la OMS, la exposición prolongada a PM10 y PM2.5 está asociada con una serie de efectos adversos para la salud, como el aumento de la mortalidad por enfermedades cardiovasculares y respiratorias, así como el agravamiento de enfermedades preexistentes. La OMS ha establecido directrices para los niveles de PM10 y PM2.5 en el aire, recomendando concentraciones anuales promedio de no más de 20 $\mu\text{g}/\text{m}^3$ para PM10 y 10 $\mu\text{g}/\text{m}^3$ para PM2.5 (Organización Mundial de la Salud, 2021).

En Ecuador, el monitoreo y control del material particulado es un desafío constante debido a factores como la urbanización rápida, el crecimiento del parque automotor, y la actividad industrial. Estudios recientes indican que varias ciudades ecuatorianas, como Quito y Guayaquil, presentan niveles de PM10 y PM2.5 que exceden las recomendaciones de la OMS, lo que plantea riesgos significativos para la salud pública (Celi Ortega, 2018).

2.1.2 Variables de contaminantes atmosféricos.

En el análisis de la calidad del aire, se prioriza la evaluación de gases nocivos, que a menudo son imperceptibles, pero tienen un impacto directo en la salud (Goshua et al., 2022). Estos gases representan una de las principales fuentes de contaminación atmosférica, con efectos significativos tanto en la salud humana como en el medio ambiente.

Entre las variables analizadas se incluyen las partículas PM10 y PM2.5, el dióxido de azufre (SO₂), el monóxido de carbono (CO), los óxidos de nitrógeno, como el NO₂, NO y NO_x, y el ozono (O₃). A continuación, se realiza una descripción más detallada de cada uno de los gases mencionados.

- NO_x (Óxidos de Nitrógeno): Son gases que incluyen el dióxido de nitrógeno (NO₂), óxido nítrico (NO), y otros compuestos de nitrógeno y oxígeno. Estos compuestos participan en los procesos atmosféricos que favorecen la generación de smog fotoquímico y la lluvia ácida. NO₂ en particular es irritante para las vías respiratorias, puede agravar el asma y reducir la función pulmonar, especialmente en niños y ancianos. Además, contribuye al cambio climático como precursor del ozono troposférico (Wang et al., 2021).
- O₃ (Ozono troposférico): Es un gas incoloro e inodoro que se forma cuando los óxidos de nitrógeno (NO_x) y compuestos orgánicos volátiles (COV) reaccionan en presencia de la luz solar (Huang et al., 2018). Aunque el ozono en la estratosfera es beneficioso, en la troposfera es un contaminante que puede causar problemas

respiratorios, dañar los pulmones, agravar enfermedades como el asma y reducir la función pulmonar (W. Hu & Yang, 2024). También afecta a las plantas, reduciendo el rendimiento agrícola y afectando a los ecosistemas naturales.

- CO (Monóxido de Carbono): Es un gas incoloro e inodoro producido por la combustión incompleta de combustibles fósiles (Goshua et al., 2022). Interfiere con la capacidad de la sangre para transportar oxígeno a través del cuerpo, lo que puede ser mortal en altas concentraciones. A niveles más bajos, puede causar síntomas como dolores de cabeza, mareos, fatiga, y en casos severos, daño cerebral. Es especialmente peligroso para personas con enfermedades cardíacas (Varon et al, 1999; Chambers et al, 2008, como se cita en Hu & Yang, 2024).
- SO₂ (Dióxido de Azufre): Es un gas incoloro con un olor irritante, producido principalmente por la quema de combustibles fósiles que contienen azufre. Puede causar irritación de los ojos, la nariz y la garganta, problemas respiratorios, y agravar enfermedades como el asma (Orellano et al., 2021). Se generan daños ecológicos y materiales mediante la lluvia ácida. Asimismo, se examinará la relación crítica entre el NO₂ y el O₃; el primero actúa no solo como contaminante directo, sino como marcador de emisiones vehiculares en zonas de alta densidad urbana.

El dióxido de nitrógeno (NO₂) y el ozono troposférico (O₃) debido a su estrecha interrelación y su papel crucial en la formación de episodios de contaminación atmosférica severa, se analizarán de manera conjunta. El NO₂, además de ser un contaminante directo, actúa como un indicador de la actividad antropogénica, especialmente en zonas urbanas con alta densidad de tráfico vehicular, que es una de las principales fuentes de emisiones de NO_x. Por otro lado, el ozono troposférico (O₃) no solo es perjudicial en sí mismo, sino que también

puede aumentar la toxicidad de otros contaminantes al reaccionar con ellos en la atmósfera (Ojeda-Castillo et al., 2024; Vazquez Santiago et al., 2024).

La elección de analizar únicamente NO₂ y O₃ permite enfocar los recursos en monitorear y comprender los procesos que más contribuyen al deterioro de la calidad del aire en áreas urbanas.

2.1.2.1 Dióxido de Nitrógeno (NO₂)

La contaminación por dióxido de nitrógeno (NO₂) es un importante problema medioambiental derivado de diversas fuentes, como las emisiones de los vehículos, los procesos industriales y la combustión. Este gas parte de los óxidos de nitrógeno (NO_x), y contribuye a la mala calidad del aire, provocando problemas respiratorios y daños medioambientales (Koziel et al., 2024). El NO₂ reacciona en la atmósfera para formar partículas nocivas y ozono, afectando a la salud humana, a los ecosistemas e incluso contribuyendo al cambio climático (Salonen et al., 2019).

En concreto las emisiones de NO₂, desempeñan un papel clave en la creación de smog fotoquímico, desencadenan la lluvia ácida y causan daños ecológicos en los embalses de agua. Además, los altos niveles de NO₂, también elevan el O₃, afectando negativamente a la agricultura. Cabe mencionar que el control y la reducción de los niveles de NO₂ son fundamentales para mitigar sus efectos adversos tanto en la salud humana como en el medio ambiente (Boldo, 2016).

2.1.2.2 Umbrales de exposición de NO₂

Los umbrales de exposición de NO₂ están regulados para proteger la salud pública y reducir los efectos adversos en el medio ambiente. Según lo establecido en las directrices mundiales de la OMS sobre la calidad del aire (2021) para los niveles de NO₂ en el aire ambiente:

- **Concentración Anual:** La OMS recomienda un límite de $10 \mu\text{g}/\text{m}^3$ para la exposición anual promedio de NO_2 .
- **Concentración de 24 Horas:** Para exposiciones más cortas, el límite recomendado es de $25 \mu\text{g}/\text{m}^3$

2.1.2.3 Ozono (O_3)

El ozono es una forma triatómica del oxígeno que se genera a partir del oxígeno atmosférico a través de procesos catalíticos influenciados por la radiación de alta energía, como la luz ultravioleta. Estas reacciones se ven notablemente aceleradas por la contaminación del aire, especialmente debido a la presencia de gases nitrosos (NO_x). La propiedad más distintiva del ozono es su capacidad excepcional para oxidar sustancias, siendo el segundo elemento más oxidante conocido (Velázquez de Castro González, 2012).

El ozono es un gas de color azul claro con un olor penetrante, que puede percibirse en concentraciones entre $19,63 \mu\text{g}/\text{m}^3$ y $98,16 \mu\text{g}/\text{m}^3$. Está formado por tres átomos de oxígeno y tiene una vida media de 40 minutos a una temperatura de 20°C (González Chamorro et al., 2024). La exposición a bajos niveles de gases oxidantes causa daño a los tejidos en el conducto de unión bronquio alveolar (Velázquez de Castro González, 2012).

La exposición a niveles bajos puede dañar los tejidos bronquio alveolares, provocando inflamación, edema y otras alteraciones celulares. Se generan radicales libres que pueden afectar la función celular a través de la peroxidación de lípidos. A concentraciones elevadas ($800 \mu\text{g}/\text{m}^3$), se observa una respuesta inflamatoria aguda y broncoconstricción. Exposiciones más breves (1-3 horas) a concentraciones entre 240 y $800 \mu\text{g}/\text{m}^3$ causan efectos restrictivos en la función pulmonar, afectando a quienes realizan actividades físicas al aire libre (Velázquez de Castro González, 2012).

2.1.2.4 Umbrales de exposición de O₃

En lo establecido por las directrices mundiales de la OMS sobre la calidad del aire (2021) para los niveles de ozono, con el fin de minimizar los riesgos para la salud y el medio ambiente se tienen:

- **Concentración en Temporada Alta:** El límite recomendado es de 60 $\mu\text{g}/\text{m}^3$ durante la temporada alta, basado en el promedio de las concentraciones máximas diarias en los seis meses consecutivos con la concentración media móvil de ozono más alta.
- **Concentración de 8 Horas:** Para exposiciones más cortas, se recomienda un límite de 100 $\mu\text{g}/\text{m}^3$ para la media de 8 horas

2.2. Internet de las cosas (IoT).

Desde el punto de perspectiva técnica la UIT-T Y.2060 (2012) define IoT como:

Una infraestructura global de la sociedad de la información. Esta infraestructura permite proporcionar servicios avanzados mediante la conexión de objetos (tanto físicos como virtuales), gracias a la interoperabilidad de las tecnologías de la información y la comunicación (TIC) actuales y futuras.

Para que un dispositivo sea catalogado como IoT debe cumplir como mínimo capacidades de comunicación, además se destaca que el auténtico valor de las aplicaciones IoT es la capacidad de analizar múltiples datos y realizar toma de decisiones basados en los mismos (Tascon & Coullaut, 2016).

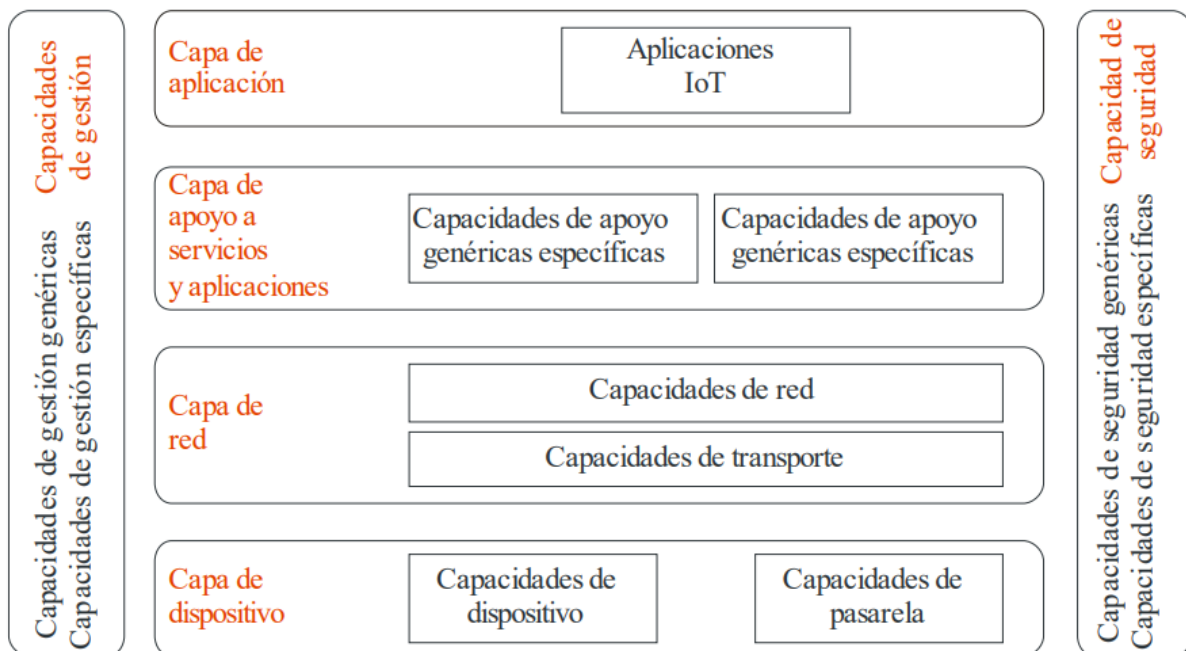
Por lo tanto, y en cuanto a su relevancia, se reconoce de manera unánime el impacto considerable y el enorme potencial disruptivo del IoT. Esta tecnología se extiende a industrias y aplicaciones cruciales, tales como la salud inteligente, el transporte inteligente, el hogar inteligente, los edificios inteligentes, la energía inteligente, la industria inteligente e incluso las ciudades inteligentes (Barrio Andrés, 2018).

2.2.1. Arquitectura de Sistemas IoT

Según las recomendaciones de UIT-T Y.2060 (2012), expone de forma clara al objetivo que cada capa posee en una arquitectura de sistemas IoT. Las capas del modelo de referencia de la arquitectura IoT son: capa de Dispositivos, capa de Red, capa para el Soporte de las Aplicaciones y Servicios (SSAS), y la capa de Aplicación como se muestra en la figura 4.

Figura 4

Arquitectura IoT



Nota. (Unión Internacional de Telecomunicaciones, 2012)

Capa de Dispositivos: La capa de dispositivo se clasifica en dos tipos de capacidades: las capacidades de dispositivo y las capacidades de pasarela. Las capacidades de dispositivo incluyen la interacción directa e indirecta con la red de comunicaciones, la construcción de redes ad-hoc, y la capacidad de cambiar entre modos de reposo y activo para ahorrar energía. Por su parte, las capacidades de pasarela permiten la conexión entre dispositivos que utilizan diferentes tecnologías y protocolos, facilitando la interoperabilidad

entre dispositivos heterogéneos (Unión Internacional de Telecomunicaciones, 2012, p. 14-15).

Capa de Red: La capa de red se compone de dos tipos de capacidades: las capacidades de red y las capacidades de transporte. Las capacidades de red se centran en el control de la conectividad, incluyendo funciones como la gestión de la movilidad y la autenticación, autorización y contabilidad (AAA). Por otro lado, las capacidades de transporte se enfocan en proporcionar conectividad para el transporte de información y datos específicos de los servicios y aplicaciones IoT, así como para la transmisión de información de control y gestión relacionada con IoT (Unión Internacional de Telecomunicaciones, 2012, p. 14).

Capa de Apoyo a Servicios y Aplicaciones (SSAS): Esta capa se divide en dos grupos de capacidades: las capacidades de soporte genéricas y las capacidades de soporte específicas. Las capacidades genéricas son comunes a múltiples aplicaciones IoT, como el procesamiento y almacenamiento de datos, mientras que las capacidades específicas están diseñadas para satisfacer las necesidades particulares de aplicaciones concretas. Estas capacidades permiten la creación de funciones de apoyo que facilitan la operación de las aplicaciones IoT (Unión Internacional de Telecomunicaciones, 2012, p. 14).

Capa de Aplicación: La capa de aplicación es la encargada de albergar las aplicaciones IoT, las cuales son responsables de ofrecer servicios específicos a los usuarios finales. Estas aplicaciones pueden variar ampliamente, desde sistemas de transporte inteligente hasta aplicaciones de ciber salud o hogares inteligentes. La capa de aplicación se enfoca en la interacción directa con el usuario y en la provisión de servicios avanzados basados en los datos recopilados y procesados por las demás capas (Unión Internacional de Telecomunicaciones, 2012, p. 14).

El modelo de referencia IoT integra capacidades transversales de gestión y seguridad. La gestión aplica el estándar FCAPS (fallos, configuración, contabilidad, rendimiento y seguridad), indispensable para asegurar la operatividad de la red y la administración de los dispositivos. Por otro lado, las capacidades de seguridad incluyen funciones como la autenticación, autorización, protección de la privacidad y control de acceso, tanto a nivel genérico como específico para aplicaciones particulares (Unión Internacional de Telecomunicaciones, 2012, p. 15).

2.2.2 Tecnologías de conectividad en IoT

Las tecnologías de conectividad que habilitan la comunicación entre dispositivos difieren en alcance, eficiencia energética y aplicaciones, lo que determina su idoneidad para diversos contextos. A continuación, se presenta un análisis comparativo de las principales tecnologías de conectividad en IoT:

- **Wi-Fi:** Opera en las bandas de 2.4 GHz y 5 GHz, con un alcance de hasta 100 metros y tasas de transmisión que oscilan entre 10 Mbps y 1 Gbps. Sin embargo, su alto consumo energético lo hace ideal para aplicaciones en hogares inteligentes y videovigilancia, donde la disponibilidad de energía es constante (IEEE Standards Association, 2023).
- **Bluetooth Low Energy (BLE):** opera en la banda de 2.4 GHz, con una tasa de datos de 1 Mbps y alcance típico de decenas de metros. Su bajo consumo lo posiciona como opción preferente para wearables y dispositivos de salud digital, como monitores cardíacos (Liu et al., 2021).
- **LoRaWAN:** Emplea frecuencias de 868 MHz (Europa) y 915 MHz (EE. UU.), destaca en aplicaciones de área extensa como la agricultura de precisión. Este último permite comunicaciones de hasta 15 km con un consumo energético mínimo, aunque con tasas de transferencia reducidas (Perera et al., 2023).

- **Zigbee:** Opera en 2.4 GHz, con alcance usa de 10 a 100 metros según el entorno y 20 - 250 kbps de velocidades típicas. Destaca en automatización industrial y sistemas de iluminación inteligente gracias a su bajo consumo y capacidad para redes en malla (Nikhade, 2015).

2.2.3. Rendimiento de los sistemas IoT en el Monitoreo de la calidad del aire.

Los sistemas de monitorización de la calidad del aire basados en IoT ofrecen datos en tiempo real con una mayor resolución espaciotemporal en comparación con los métodos tradicionales (Buelvas et al., 2023). Estos sistemas suelen medir parámetros como la temperatura, la humedad, el CO, el CO₂ y las partículas (Firdaus et al., 2019; Kumar & Jasuja, 2017). La calibración de los sensores es crucial para la precisión, ya que los sensores calibrados muestran un error de medición del 4,3 % frente al 8,5 % de los no calibrados. Factores ambientales como la temperatura y la humedad afectan al rendimiento de los sensores, mientras que la ubicación de los sensores influye significativamente en los datos sobre la calidad del aire (Blinova et al., 2024). Se emplean varios métodos de transmisión de datos, como Wi-Fi, redes celulares y LPWAN LoRa, con Wi-Fi mostrando menor latencia y pérdida de datos (Blinova et al., 2024; Firdaus et al., 2019). Las plataformas de computación en la nube como IBM Bluemix y Antares se utilizan para el almacenamiento y la visualización de datos (Firdaus et al., 2019; Kumar & Jasuja, 2017). Sin embargo, persiste la preocupación por la calidad de los datos, siendo la exactitud y la precisión las dimensiones más comunes que se abordan mediante técnicas de calibración (Buelvas et al., 2023).

2.3. Redes Neuronales Artificiales (ANN).

Gracias a su capacidad para modelar datos sin reglas explícitas, las ANN se han consolidado como una herramienta clave en la minería de datos. Su funcionamiento depende

de unidades de procesamiento que aplican funciones de activación a entradas ponderadas, perfeccionando el modelo mediante iteraciones constantes de ajuste de pesos (Chollet, 2021).

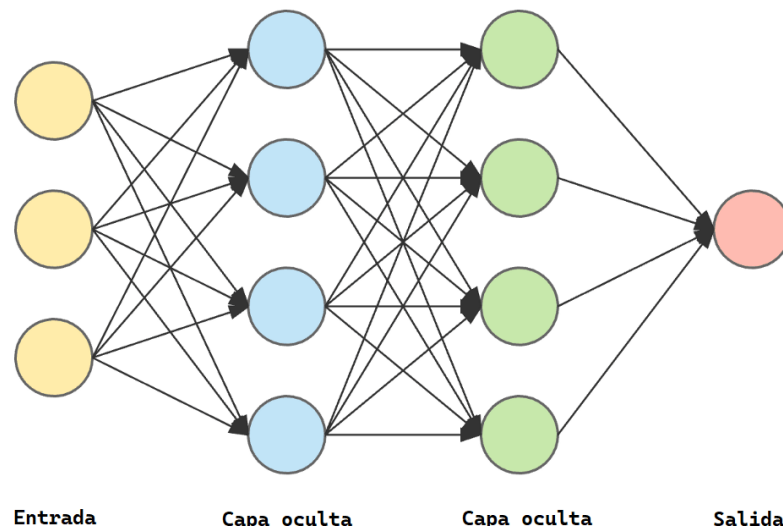
La implementación de redes neuronales a basadas en datos para sistemas de calidad del aire en arquitecturas IoT suele producirse en varias capas. En la capa de borde, los modelos ligeros de ML, incluidas las ANN, pueden desplegarse en dispositivos de bajo coste para un procesamiento eficiente de los datos cerca de la fuente (Gamazo-Real et al., 2022). La capa de detección recopila datos de calidad del aire utilizando sensores terrestres y aéreos, mientras que la capa de transmisión permite la comunicación bidireccional (Z. Hu et al., 2019). El procesamiento y el análisis de los datos se producen en la capa de procesamiento, donde se pueden aplicar técnicas de ajuste espacial y predicción a corto plazo (Z. Hu et al., 2019). Los algoritmos de aprendizaje profundo para la detección de anomalías se pueden implementar en varias capas de IoT, incluidas las capas de red y aplicación (Astaras et al., 2019).

2.3.1. Estructura y funcionamiento de las Redes Neuronales

En una red neuronal artificial, la distribución de las neuronas se organiza en niveles, cada uno compuesto por un número determinado de neuronas. Cuando un conjunto de neuronas recibe simultáneamente el mismo tipo de información, se denomina *capa*. Generalmente, se distinguen tres capas: de entrada, oculta y de salida. Cada neurona recibe una serie de entradas provenientes de la salida de otras neuronas de la red. Una vez calculada la salida de una neurona en la capa oculta, esta se propaga, a través de las conexiones de salida, hacia las neuronas destino, recibiendo todas ellas el mismo valor de salida (Montesinos López et al., 2022).

Figura 5

Estructura Básica de una red neuronal



En una red neuronal, podemos diferenciar tres tipos de niveles:

- **Entrada:** Es el conjunto de neuronas que recibe directamente la información proveniente de fuentes externas a la red.
- **Oculto:** Corresponde a un conjunto de neuronas internas a la red que no tienen contacto directo con el exterior. El número de niveles ocultos puede variar desde cero hasta un número elevado. Generalmente, las neuronas de cada nivel oculto comparten el mismo tipo de información, por lo que se denominan formalmente capas ocultas. Las neuronas de estas capas pueden estar interconectadas de diversas maneras, lo que, junto con su número, determina las distintas arquitecturas de redes neuronales (Goodfellow et al., 2016).
- **Salida:** Es el conjunto de neuronas que transfiere la información procesada por la red hacia el exterior.

2.3.2. Tipos de redes Neuronales.

Existen varios tipos de redes neuronales artificiales dependiendo el modelo de la arquitectura, cada una con características y aplicaciones específicas. De ese modo tenemos los siguientes tipos de modelos:

2.3.2.1. Redes Neuronales Feedforward (FNN).

Una FNN es una arquitectura de red neuronal en la que la información fluye en una sola dirección, desde la entrada hasta la salida, sin ciclos ni bucles de retroalimentación. Esto las diferencia de las redes recurrentes (RNN), que procesan información secuencial gracias a sus conexiones cíclicas (Wilamowski & Irwin, 2018). Los tipos de FNN más utilizados son:

- Perceptrón multicapa (MPL): Utiliza una arquitectura de capas (entrada, ocultas y salida), donde cada neurona se conecta mediante pesos ajustables. Se implementan funciones de activación como *ReLU* o variantes para mejorar el aprendizaje de relaciones no lineales. Además su entrenamiento combina retropropagación con optimizadores como Adam (Kingma & Ba, 2014).
- Red neuronal de regresión generalizada (GRNN): El modelo ha tenido mejoras continuamente desde su origen en los 90, como es la integración con técnicas de regularización para evitar sobreajuste en problemas de regresión (Martínez et al., 2022). Su estructura de cuatro capas sigue siendo útil en aplicaciones que requieren aproximación rápida de funciones.
- Red de función de base radial (RBF): Las redes RBF han evolucionado con enfoques híbridos, como su combinación con capas convolucionales para procesamiento de imágenes (Amirian & Schwenker, 2020). Su capa oculta transforma no linealmente los datos, mientras la capa de salida aplica modelos lineales adaptativos.

2.3.2.2. Redes Neuronales Recurrentes (RNN).

Son un tipo de red neuronal artificial diseñada para procesar datos secuenciales. A diferencia de las redes *feedforward*, las RNN pueden retener información pasada gracias a conexiones recurrentes, lo que les permite analizar secuencias como texto o series temporales (Mienye et al., 2024). Los tipos de RNN más utilizados son:

- Red de Elman: Esta arquitectura simple incluye una capa de contexto que almacena estados anteriores, permitiendo "memoria" para tareas dependientes del orden, como modelado de series temporales. Su estructura consta de tres capas: entrada, oculta (con retroalimentación a través de la capa de contexto) y salida. Aunque clásica, se han propuesto mejoras recientes en su estabilidad para secuencias largas (Salehinejad et al., 2017).
- LSTM: Diseñadas para resolver el desvanecimiento/explosión de gradientes en RNN tradicionales, las LSTM usan células de memoria con compuertas (entrada, salida y olvido) para gestionar información a largo plazo. La evolución de estos modelos desde 1997, impulsada por mecanismos de atención, les permite competir en traducción y generación de texto (Vaswani et al., 2017). Pese a la aparición de los transformadores, continúan siendo una herramienta eficaz y relevante para tareas lingüísticas (Sherstinsky, 2020).

2.3.2.3. Redes Neuronales Híbridas (HNN).

Las redes neuronales híbridas representan una evolución en el aprendizaje profundo, combinando arquitecturas diversas para resolver problemas complejos con mayor eficacia. Estas redes fusionan técnicas y modelos, aprovechando sus fortalezas individuales para crear sistemas robustos y adaptativos (R. Zhao et al., 2022). Los tipos de HNN más utilizados son:

- Datos – Intensivo: Diseñadas para integrar grandes volúmenes de datos heterogéneos (estructurados, imágenes, texto, series temporales), estas redes combinan arquitecturas como CNNs y RNNs para procesar múltiples fuentes y mejorar la capacidad predictiva. Se utilizan sistemas multimodales para unir información visual y textual en tareas descriptivas (F. Zhao et al., 2024).
- Modelo – Intensivo: la integración de modelos supervisados y no supervisados, como arquitecturas CNNs para imágenes y LSTMs para

secuencias, logrando sistemas más precisos que permite procesar patrones espaciotemporales complejos con una mayor exactitud (Alomar et al., 2024).

- Técnicas – Intensivo: Emplean métodos avanzados como regularización dinámica, meta-aprendizaje y optimización evolutiva para adaptarse a distintos datos y tareas. Por ejemplo, técnicas de *neural architecture search* (NAS) automatizan el diseño de arquitecturas híbridas, optimizando su rendimiento en tiempo real (Mecharbat et al., 2023).

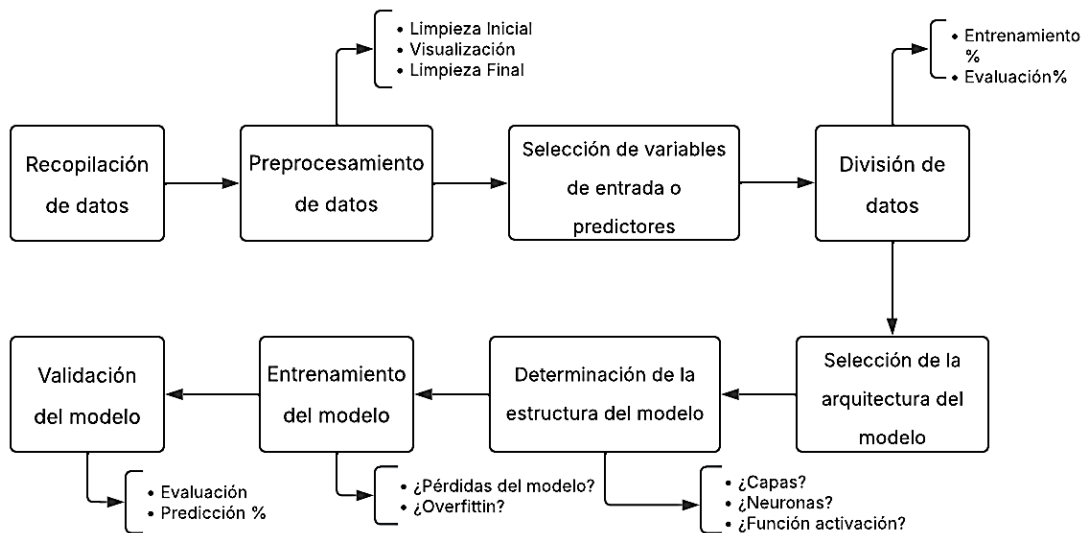
2.3.3. Proceso de Desarrollo de ANN

En los últimos años, las redes neuronales se han implementado con éxito en diferentes aplicaciones de predicción a corto y largo plazo, sin embargo, debido a que cada modelo de red neuronal es único para cada aplicación, no existe un método establecido para la construcción (Cabaneros et al., 2019).

No obstante, en la literatura existen diversas directrices que ofrecen un enfoque sistemático para desarrollar modelos de redes neuronales artificiales. El proceso se puede dividir en 8 pasos principales: recopilación de datos, preprocesamiento de datos, selección de variables de entrada, división de datos, selección de la arquitectura del modelo, determinación de la estructura del modelo, entrenamiento del modelo y validación del modelo. En la figura 6 se presenta un diagrama que resume el proceso de creación de una red neuronal artificial (ANN).

Figura 6

Diagrama de procedimiento de ANN



Nota. Autoría Propia

2.3.1.1. Recopilación de datos

La recopilación de datos es un aspecto crítico debido a que requiere una planificación cuidadosa para determinar el propósito de la red, los requisitos de datos y los resultados deseados. Una selección adecuada de datos mejora el rendimiento del algoritmo y reduce costes computacionales en problemas de clasificación y regresión (Ferreira et al., 2021).

Para garantizar una recopilación de datos eficaz, es fundamental implementar técnicas de limpieza, eliminación de valores atípicos, el manejo de datos faltantes y la normalización de las variables (García et al., 2015).

2.3.1.2. Preprocesamiento de datos

El preprocesamiento de datos es crucial para un entrenamiento y un rendimiento eficaz, ya que se pretende mejorar la representación de los datos recogidos. Las dos técnicas populares de preprocesamiento de datos en el campo de la modelización de la contaminación atmosférica son la normalización y la imputación de datos (Cabaneros et al., 2019).

Para eliminar disparidades de escala y cumplir con los requisitos de la capa oculta, se aplican métodos de normalización. Estos incluyen el escalado de valores (ej. 0 a 1) y la estandarización para obtener una distribución con media cero (Cabaneros et al., 2019).

Para manejar los vacíos de información sin distorsionar el modelo, se descartan soluciones básicas como el promedio. En cambio, se priorizan métodos avanzados de imputación múltiple o machine learning, asegurando una reconstrucción precisa de los datos faltantes (Hua et al., 2024).

2.3.1.3. Selección de variables de entrada o predictores

La solidez de un modelo depende en gran medida de cómo se seleccionan e introducen los predictores. Incluir predictores irrelevantes o correlacionados puede causar sobreajuste, mientras que la falta de variables explicativas importantes impide que el modelo capture adecuadamente la dinámica subyacente (Cabaneros et al., 2019). Existen dos métodos principales para seleccionar predictores significativos: enfoques sin modelo (como el método ad-hoc y el método analítico) y enfoques basados en modelos. Sin embargo, los enfoques basados en modelos son lentos y no miden claramente el efecto de cada predictor debido a la influencia de múltiples parámetros de la red (Cabaneros et al., 2019).

2.3.1.4. División de datos

La división de datos es un paso esencial en el desarrollo de modelos de redes neuronales. Se realiza dividiendo los datos disponibles en tres conjuntos: entrenamiento, validación y prueba. El conjunto de entrenamiento se usa para calcular el gradiente y ajustar los pesos y sesgos de la red. Para evitar el sobreajuste, el sistema monitoriza el error de validación y finaliza el entrenamiento al detectar incrementos, reteniendo los pesos más precisos. El proceso concluye usando el conjunto de prueba para medir la generalización real (Cabaneros et al., 2019).

Los datos se dividen en conjuntos de entrenamiento, validación y prueba, ya sea mediante métodos supervisados o no supervisados. En cambio, los métodos no supervisados no consideran explícitamente estas propiedades, aunque los enfoques estratificados intentan que las propiedades estadísticas sean similares en cada conjunto (Maier et al., 2010).

2.3.1.5. Selección de la arquitectura del modelo

En la selección de arquitectura se refiere a la estructura del modelo se refiere a la estructura general y a la forma en que la información fluye de una capa a otra en las redes neuronales.

La arquitectura se determina mediante métodos como ensayo y error o técnicas de optimización (Aladag, 2019). La eficiencia de la red depende de una adecuada selección de variables, capaz de eliminar hasta el 85% de la información redundante (Yousaf et al., 2017). Herramientas como el análisis de regresión son esenciales en esta etapa para definir la mejor configuración y verificar la precisión (Aladag, 2019).

2.3.1.6. Determinación de la estructura del modelo

La determinación de la estructura es establecer el número de capas y nodos en cada capa, dependiendo de diversos factores como la naturaleza del problema a resolver, la cantidad y calidad de los datos disponibles, y la capacidad de cómputo.

En este contexto, la capa de entrada procesa los predictores, la capa oculta modela las relaciones complejas entre dichos predictores y la variable objetivo, y la capa de salida genera el resultado final (Lecun et al., 2015). Un número adecuado de nodos en la capa oculta permite aproximar funciones complejas, aunque un exceso puede inducir sobreajuste, mientras que muy pocos nodos pueden provocar sub-ajuste (Zhang et al., 2016). Además, diversas metodologías —desde enfoques globales e iterativos hasta estrategias basadas en la experiencia— se emplean para establecer la estructura óptima del modelo, adaptándose a la aplicación específica (Lecun et al., 2015).

La estructura óptima se determina mediante métodos globales, iterativos o basados en experiencia (Cabaneros et al., 2019).

2.3.1.7. Entrenamiento del modelo

El procedimiento de ajuste de las neuronas de una red es llamado Procedimiento de Aprendizaje o Procedimiento de Entrenamiento. Los entrenamientos de redes se dividen supervisados y no supervisados.

En el entrenamiento supervisado, se usan datos de entrada y salida esperada para ajustar el modelo. Como su mismo termino los procedimientos de entrenamiento utilizan un parámetro para supervisar el desempeño de la red. Este parámetro se emplea para determinar las mejores modificaciones con el fin de mejorar el rendimiento. En las redes supervisadas, el desempeño generalmente se mide numéricamente mediante la magnitud del error de aprendizaje (Perez, 2012).

Las redes neuronales cuyas arquitecturas asumen que el conjunto de entrenamiento solo consta de vectores de entrada se conocen como Redes Neuronales No Supervisadas. En estas redes, no se puede medir numéricamente la magnitud del error de aprendizaje. En cambio, el entrenamiento se enfoca en identificar la estructura y las similitudes entre los vectores de entrada. Para clasificar un vector, el análisis de similitud puede basarse en la evaluación y comparación de características topológicas o geométricas (Perez, 2012).

2.3.1.8. Validación del modelo

La validación evalúa el rendimiento del modelo en datos no vistos para garantizar su precisión. Un marco de validación integral debe abarcar la validez replicativa, estructural y predictiva para mitigar problemas como la equifinalidad, donde distintos parámetros arrojan resultados similares (Humphrey et al., 2017). Este enfoque evalúa la idoneidad de la arquitectura (Elsken et al., 2018) y se distingue de la selección de modelos: mientras esta última minimiza el desajuste, la validación se centra en reducir los errores de predicción (Bengio, 2012).

La evaluación de estos modelos abarca la validez replicativa, que asegura la coherencia con los datos de entrenamiento (Gulshan et al., 2016); la predictiva, enfocada en la generalización con datos independientes; y la estructural, que verifica si el modelo captura correctamente las relaciones internas de los datos (Wu et al., 2014).

2.3.4. Tipos de redes neuronales más usadas en sistemas de calidad del aire.

Las redes feedforward y los modelos híbridos son populares para tareas de predicción. Dentro del primer grupo, los perceptrones multicapa (MLP) son ampliamente utilizados para pronosticar y controlar la calidad del aire (Cabaneros et al., 2019).

Para capturar patrones espaciotemporales con alta eficiencia, Alléon et al. (2020) ilustran esta aplicación con un sistema predictivo que integra capas convolucionales y LSTM. Este diseño híbrido destaca por su eficiencia computacional, logrando un equilibrio entre la simplicidad de las redes feedforward y la capacidad para procesar patrones espaciotemporales complejos.

Su capacidad para modelar interacciones complejas, combinada con una estructura simple y flexible, las posiciona como una opción preferente para el cálculo de concentraciones de contaminantes (Cabaneros et al., 2019).

Por otro lado, en sistemas de calidad del aire también se utilizan redes neuronales recurrentes, como las LSTM, y modelos híbridos basados en enfoques data-intensive. Las LSTM son valoradas por su capacidad para procesar secuencias de datos y aprender dependencias a largo plazo, lo cual es crucial en escenarios donde la temporalidad y el orden de los eventos influyen significativamente en la precisión del pronóstico (Belavadi et al., 2020). Para abordar la complejidad de la contaminación atmosférica, estos sistemas integran múltiples métodos de aprendizaje. Al clasificar los datos según sus comportamientos, generan modelos especializados de alta precisión (Aggarwal & Toshniwal, 2021).

Ambas arquitecturas ofrecen ventajas específicas que las hacen adecuadas para abordar problemas complejos en la predicción y el control de la calidad del aire. En la tabla 1 se mencionan algunos trabajos similares donde se aplicó modelos de redes neuronales con el mismo objetivo del monitoreo de calidad del aire y parámetros ambientales.

Tabla 1

Aplicación de ANN en el Monitoreo y Predicción de la Calidad del Aire

Estudio	Autor y año	Tipo de ANN / Modelo	Resultados principales
Monitoreo de la calidad del aire en un espacio concurrido aplicando internet de las cosas y redes neuronales artificiales	Alejandro Benavidez & Millán Cruz (2023)	Perceptrón Multicapa (MLP) feedforward	Se evidenció la capacidad de identificar variaciones en la calidad del aire en función de condiciones ambientales.
Artificial neural network an innovative approach in air pollutant prediction for environmental applications: A review.	Yadav et al. (2024)	Modelo Perceptrón Multicapa (MLP), Redes Neuronales Convolucionales (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit	Se evidenció que los modelos basados en ANN y deep learning superan a los métodos tradicionales en precisión de predicción, gracias a su capacidad para manejar relaciones no

		(GRU), Modelos híbridos	lineales y múltiples variable
Red Neuronal Artificial aplicado para el pronóstico de eventos críticos de PM2.5 en el Valle de Aburrá	Baena-Salazar et al. (2019)	Perceptrón Multicapa (MLP) feedforward	Capacidad para predecir eventos críticos de contaminación con eficiencia superior al 70% y margen de error reducido.
LSTM-Autoencoder based Anomaly Detection for Indoor Air Quality Time Series Data	Wei et al. (2022)	Modelo híbrido basado en LSTM-Autoencoder	Alcanzó una precisión del 99.50% en la detección de anomalías, permitiendo identificar posibles fallos o eventos críticos en ambientes interiores.
Blind Calibration of Air Quality Wireless Sensor Networks Using Deep Neural Networks	Veiga et al. (2021)	Deep Neural Network (DNN) – modelo de red profunda de retropropagación	Reducción del error de calibración en un orden de magnitud frente a métodos tradicionales.

Cada estudio presenta una aproximación adaptada al problema y a la disponibilidad de datos, utilizando desde arquitecturas clásicas de MLP hasta enfoques híbridos como LSTM-Autoencoder, lo que evidencia la versatilidad de las RNA en aplicaciones ambientales y de monitoreo de la calidad del aire.

CAPÍTULO III: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

En este presente capítulo se aborda el diseño y desarrollo del sistema de monitoreo de calidad de aire para concentraciones de dióxido de nitrógeno (NO₂) y ozono (O₃). De acuerdo con la metodología planteada, se utiliza un enfoque bottom-up para iniciar con el análisis de requerimientos que debe contar el sistema, el cual constituye la base para la selección del hardware y software más adecuados para su implementación. Este proceso se alinea con el estándar IEEE 29148, el cual guía la especificación y gestión de requisitos del sistema. Posteriormente, se describe el proceso de construcción del sistema para luego llevar a cabo pruebas de funcionamiento.

3.1. Metodología

La metodología utilizada en el desarrollo del sistema es Bottom-Up, la cual se divide principalmente en 4 fases principales que se unifican hasta obtener un sistema final. A continuación, se da detalla brevemente cada una de las fases.

- **Fase de especificaciones:** Se lleva a cabo una investigación bibliográfica para establecer una base sólida en el desarrollo del sistema.
- **Fase de verificación y simulación:** Se realiza la definición de los requerimientos de hardware y software, los cuales son fundamentales para posteriormente la selección de los componentes y el diseño de la arquitectura del sistema.
- **Fase de integración de bloques:** Se ejecuta la construcción e integración de los componentes con el objetivo de conformar el sistema final.
- **Fase de verificación del sistema:** Se efectúan las pruebas de funcionamiento correspondientes para la validación del sistema.

3.2. Análisis de la situación actual.

En la actualidad, no existe información actualizada sobre la calidad del aire en la ciudad de Ibarra, debido a que, a partir de 2016, el daño de los equipos de monitoreo y la falta de continuidad en los convenios interinstitucionales han impedido un seguimiento sistemático (GAD Municipal de San Miguel de Ibarra, 2022). Esta situación limita la capacidad de evaluar el impacto de fuentes contaminantes como el parque automotor, la actividad industrial y la quema de desechos, afectando la planificación ambiental y la toma de decisiones estratégicas.

Ante esta problemática, se propone desarrollar un sistema IoT de monitoreo de calidad del aire para la medición de dióxido de nitrógeno (NO₂) y ozono (O₃), utilizando redes neuronales para mejorar la recopilación y procesamiento de los datos obtenidos por los sensores.

El sistema permitirá visualizar mediante una interfaz web la información en tiempo real y de forma accesible, buscando determinar si el estado actual de la calidad del aire en la ciudad de Ibarra, referente a las concentraciones de los contaminantes mencionados, cumple con los estándares ambientales establecidos.

3.3. Requerimientos del sistema.

En el siguiente apartado, de acuerdo con la fase de verificación y simulación de la metodología implementada, se detallan los requerimientos necesarios para la construcción del sistema. Este proceso se realiza en concordancia con el estándar IEEE 29148, que guía la especificación y gestión de requisitos. Se contemplan los requerimientos de los stakeholders, así como los requerimientos del sistema y de la arquitectura.

3.3.1. Nomenclatura de requerimientos

La Tabla 2 muestra las abreviaturas de los requerimientos de stakeholders, del sistema y de arquitectura, que se utilizan posteriormente.

Tabla 2*Nomenclatura de Requerimientos*

Abreviatura	Significado
StRS	Requerimientos de Stakeholders
SyRS	Requerimientos del Sistema
SrSH	Requerimientos de Arquitectura

Por otra parte, se llevó a cabo una clasificación de los requerimientos en niveles de alta, media y baja prioridad. Este enfoque permitió identificar de forma precisa aquellas funciones críticas que deben operar sin errores y distinguir las que pueden ser opcionales sin comprometer el rendimiento global del dispositivo.

3.3.2. Identificación de Stakeholders

En este proyecto, los stakeholders participan de manera activa en todas las fases, asumiendo diversas responsabilidades en las distintas áreas del sistema para garantizar que la solución final sea la más idónea. La Tabla 3 se detalla la participación de quienes contribuirán al desarrollo de este proyecto.

Tabla 3*Lista de participantes en el proyecto*

Stakeholders	Función
Negrete Ramírez Michael Israel	Desarrollador del proyecto
MSc. Cuzme Rodríguez Fabián Geovanny	Director del proyecto
MSc. Michilena Calderón Jaime Roberto	Asesor del proyecto

3.3.3. Requerimientos de Stakeholders

Una vez identificados los participantes en el desarrollo del proyecto, la Tabla 4 presenta de manera detallada los requerimientos de los stakeholders, establecidos en función

de las necesidades operativas y de usuario del sistema. Es imperativo que el sistema satisfaga de manera integral cada uno de los requisitos establecidos, ya que esto constituye un factor determinante para asegurar su óptimo funcionamiento y la consecución de los objetivos propuestos. Los requisitos especificados en el **Anexo A** permiten una comprensión más clara y exhaustiva de las necesidades y expectativas de las partes interesadas con respecto al sistema.

Tabla 4

Requerimientos de Stakeholders

Requerimientos de Stakeholders					
No.	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
Requerimientos operacionales					
StRS1	El sistema debe almacenar los datos de NO ₂ y O ₃ .	x			SyRS2
StRS2	El sistema debe proporcionar los niveles de concentraciones de contaminantes en el aire.	x			SyRS3
StRS3	El sistema debe almacenar coordenadas GPS de las mediciones realizadas.		x		
StRS4	El sistema debe almacenar datos de temperatura y humedad.		x		
StRS5	El sistema debe tener conexión inalámbrica para la transmisión de datos.	x			SrSH8
StRS6	El sistema debe tener alimentación por medio de una batería.		x		SyRS6

Requerimientos de Usuario

StRS7	La interfaz web debe ser intuitiva y fácil de usar.	x		SyRS7
StRS8	La interfaz web debe mostrar el estado de concentraciones de NO2 y O3.	x		SyRS8
StRS9	Portabilidad para usarse en distintas ubicaciones		x	SrSH5

3.3.4. Requerimientos del sistema.

Los requerimientos del sistema especifican de manera detallada los servicios que este debe proporcionar, así como las restricciones y limitaciones que deben ser cumplidas con rigurosidad, en función de su nivel de prioridad. En la tabla 5 se organizan y clasifican en distintas secciones cada uno de los requerimientos.

Tabla 5

Requerimientos del sistema.

Requerimientos del sistema					
No.	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
Requerimientos Funcionales					
SyRS1	El sistema debe ser capaz de procesar y transmitir datos en tiempo real.	x			
SyRS2	El sistema debe tener una detección adecuada de los NO2 y O3 para su análisis.	x			StRS1

SyRS3	El sistema debe implementar una red neuronal para detección y predicción de niveles de contaminación del aire	x	StRS2
-------	---	---	-------

Requerimientos de Interfaz

SyRS4	La placa de procesamiento debe incluir pines de comunicación para el intercambio de datos.	x	
SyRS5	La placa de procesamiento debe contar con pines I2C.	x	
SyRS6	Alimentación eléctrica mediante el puerto USB o conexión de 3.3-5V	x	StRS6

Requerimientos de uso

SyRS7	La interfaz web debe ser intuitiva para la interacción del usuario.	x	StRS7
SyRS8	La interfaz web debe mostrar un indicador para el estado de la detección de los contaminantes.		x StRS8
SyRS9	Facilidad de manejo y entendimiento.	x	

3.3.5. Requerimientos de arquitectura

En la tabla 6 se muestra los elementos fundamentales que conformaran el sistema, lo que permitirá una selección eficiente y adecuada de los componentes de hardware y software.

Tabla 6

Requerimientos de Arquitectura

Requerimientos de arquitectura SrSH					
No.	Requerimientos	Prioridad			Relación
		Alta	Media	Baja	
Requerimientos Lógicos					
SrSH1	La placa de procesamiento debe ser compatible con los sensores.	x			
SrSH2	Compatibilidad con tecnologías de conexión a internet.		x		StRS5
SrSH3	Disponibilidad de librerías para la funcionalidad de los sensores y placa de procesamiento.		x		
SrSH4	El lenguaje de programación debe ser compatible con los sensores.	x			
Requerimientos de Diseño					
SrSH5	El sistema debe contar con un diseño compacto que garantice su portabilidad.		x		StRS9
SrSH6	El diseño debe permitir a los sensores tener una adecuada recepción de las variables ambientales.	x			
Requerimientos de Hardware					

SrSH7	La placa de procesamiento debe proporcionar suficiente capacidad de procesamiento.	x	
SrSH8	La placa de procesamiento debe contar con tecnología de comunicación inalámbrica.	x	StRS5
SrSH9	Los sensores deben ser compatibles con la placa de procesamiento	x	
SrSH10	Los sensores deben trabajar con voltajes de hasta 5v para eficiencia en el consumo energético.	x	
SrSH11	Los sensores deben operar en exteriores.		x
SrSH12	Tarjeta de memoria microSD mínimo 64 Gb.	x	
SrSH13	Fuente de alimentación de 5v.	x	StRS6
Requerimientos de Software			
SrSH14	El IDE o lenguaje de programación deben ser de código abierto.	x	
SrSH15	Software fácil de usar y que no requiera una curva de aprendizaje extensa		x
SrSH16	Compatibilidad de librerías para el uso algoritmos de inteligencia artificial.	x	SyRS3

3.4. Selección de Hardware y Software del sistema.

A continuación, se realiza la selección de hardware y software del sistema tomando como punto de referencia los stakeholders definidos anteriormente. Para la selección se utiliza tablas comparativas y una valoración de cumplimiento de requisitos siendo seleccionada la de mayor afinidad al proyecto.

3.4.1. Selección de Hardware

A continuación, se presenta la elección de placa de procesamiento, sensores y baterías del sistema.

3.4.1.1. Microcontrolador.

Para la selección del microcontrolador destinado a la lectura de sensores, se tomaron en cuenta tanto los requerimientos técnicos y funcionales del proyecto como las necesidades de los stakeholders involucrados. En este contexto, se evaluaron tres alternativas principales: el módulo Arduino Mega Mini 250 Pro, el Arduino Uno y el Arduino Nano. La Tabla 7 presenta un análisis comparativo detallado de estas opciones, considerando criterios clave para su aplicación en el sistema propuesto.

Tabla 7

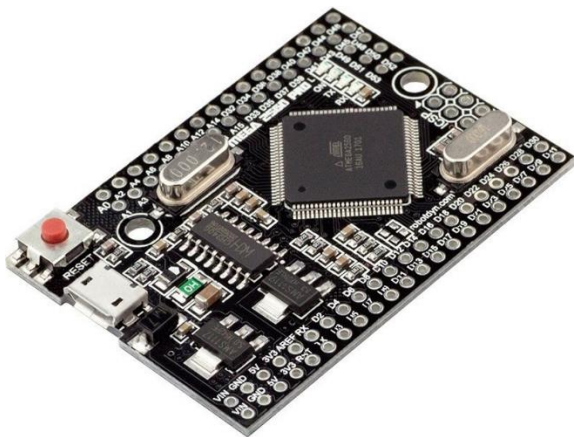
Selección del microcontrolador

Hardware	Requerimientos						Valoración
	SyRS4	SyRS5	SyRS6	SrSH1	SrSH3	SrSH7	Total
Arduino Mega Mini 250 Pro	1	1	1	1	1	1	7
Arduino UNO	1	1	1	1	1	0	6
Arduino NANO	1	1	1	1	1	0	6

De acuerdo con los resultados de la evaluación presentada en la Tabla 7, el microcontrolador Arduino Mega 250 Mini Pro (ver Figura 7) se identifica como la opción más adecuada para la implementación del sistema. Esta elección se justifica por su elevada cantidad de pines de entrada/salida, su mayor capacidad de memoria y procesamiento, así como por su formato compacto y versatilidad en la alimentación. Estas características permiten una integración eficiente con múltiples sensores y garantizan la escalabilidad y estabilidad del sistema.

Figura 7

Arduino MEGA Mini 250 Pro



Nota. (DFRobot, 2025)

3.4.1.2. Placa de procesamiento

Para la selección de la placa de procesamiento, se llevó a cabo una evaluación exhaustiva considerando los stakeholders previamente establecidos, así como los requisitos técnicos y funcionales del proyecto. Se analizaron tres alternativas principales: el módulo ESP32-WROOM-32, la placa Arduino Nano RP2040 Connect y la Raspberry Pi Zero 2W. En la Tabla 8 se presenta un análisis comparativo de las opciones.

Tabla 8

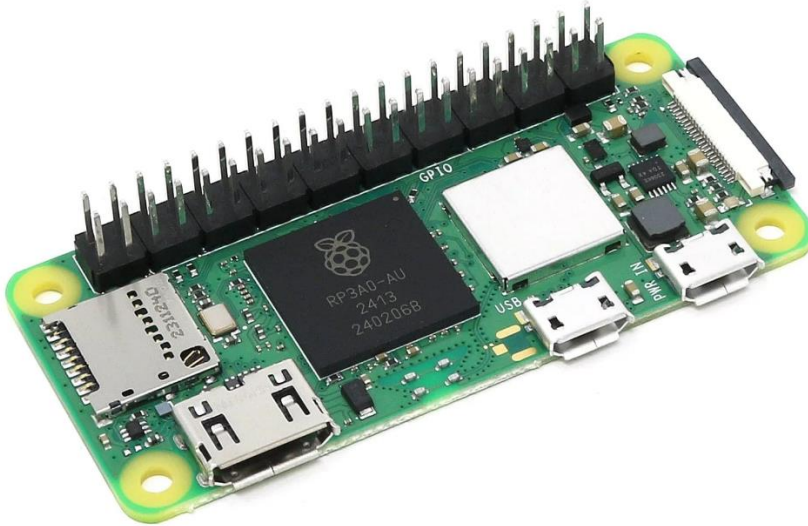
Selección de la placa de procesamiento

Hardware	Requerimientos							Valoración
	StRS5	SyRS4	SyRS5	SyRS6	SrSH1	SrSH2	SrSH8	Total
ESP32- WROOM-32	1	1	1	0	0	1	1	5
Arduino Nano RP2040	1	1	1	1	1	1	0	6
Connect Raspberry Pi Zero 2W	1	1	1	1	1	1	1	7

Conforme a los resultados de la Tabla 8, la Raspberry Pi Zero 2W (Figura 8) constituye la alternativa óptima. Su elección se justifica por la amplia compatibilidad con periféricos y el soporte para Python, lo cual simplifica significativamente el desarrollo y la implementación del sistema. Estas características no solo garantizan un desarrollo eficiente del sistema, sino que también facilitan su escalabilidad y mantenimiento a futuro permiten una alimentación estable, tanto desde USB como desde fuentes externas.

Figura 8

Raspberry Pi Zero 2W



Nota. (Seeed Studio, 2025)

3.4.1.3. Selección de Sensores

El sistema integrará un conjunto de sensores diseñados para medir variables ambientales clave, incluyendo dióxido de nitrógeno (NO₂), ozono (O₃), temperatura, humedad relativa y un módulo GPS para geolocalizar los puntos de recolección de datos. La selección de estos componentes se llevó a cabo mediante un análisis comparativo basado en criterios fundamentales como la disponibilidad en el mercado, la accesibilidad en términos de costos y la fiabilidad de las mediciones proporcionadas.

- **Sensor de NO₂**

El sensor de NO₂ mide la concentración de dióxido de nitrógeno en el ambiente, siendo relevante para el desarrollo del dispositivo debido a que permite monitorear la calidad del aire y detectar niveles potencialmente peligrosos de este gas. En la tabla 9 se comparan y evalúan los sensores NO₂ disponibles para su implementación, tomando en cuenta los requerimientos establecidos para el sistema.

Tabla 9

Selección de sensor NO₂

Hardware	Requerimientos	Valoración
----------	----------------	------------

	StSR1	StSR2	SyRS2	SrSH9	SrSH10	SrSH11	
Gravity: Factory	1	1	1	1	1	1	6
Calibrated							
Electrochemical NO2							
Winsen ZE03-NO2	1	0	1	0	1	0	3
Alphasense NO2- B43F	1	1	1	1	1	0	5

Con los resultados de la tabla 9 se concluye que el sensor ideal para la medición de dióxido de nitrógeno (NO₂) en el dispositivo será el Gravity: Factory Calibrated Electrochemical Nitrogen Dioxide Sensor (0-20ppm, I2C&UART) (Ver Figura 9). Gracias a su calibración de fábrica, el dispositivo mide con exactitud niveles de NO₂ de 0 a 20 ppm. La conectividad mediante puertos I2C y UART simplifica su acoplamiento al hardware, permitiendo una lectura de datos ágil y estable.

Figura 9

Gravity: Factory Calibrated Electrochemical Nitrogen Dioxide Sensor



Nota. (DFRobot, 2025)

- **Sensor de O₃.**

El sensor de ozono (O3) es un componente clave en el diseño del dispositivo, ya que permite cuantificar la presencia de este gas en la atmósfera. Su función es crucial para evaluar la calidad del aire, especialmente en entornos donde el ozono puede alcanzar niveles perjudiciales para la salud humana y el medio ambiente. En la tabla 10 se comparan y evalúan los sensores de O3 disponibles para su implementación, tomando en cuenta los requerimientos establecidos para el sistema

Tabla 10

Selección de sensor O3

Hardware	Requerimientos						Valoración
	StSR1	StSR2	SyRS2	SrSH9	SrSH10	SrSH11	
Gravity:	1	1	1	1	1	1	6
Electrochemical Ozone Sensor (O3)							
Winsen ZE03-O3	1	0	1	1	1	0	4
Alphasense O3-B43F	1	1	1	1	1	0	5

De acuerdo con la evaluación presentada en la Tabla 10, se determina que el sensor más adecuado para la detección de ozono (O3) en el dispositivo es el Gravity: Electrochemical Ozone Sensor (O3) (Ver figura 10). Este sensor destaca por operar en un rango de medición de 0 a 10 ppm, lo que lo hace idóneo para registrar variaciones precisas en la concentración de ozono. Cuenta con una comunicación I2C por lo que facilita una comunicación rápida para el procesamiento en tiempo real de la información obtenida.

Figura 10

Gravity: Electrochemical Ozone Sensor



Nota. (DFRobot, 2025)

- **Sensor de temperatura y humedad.**

El monitoreo de temperatura y humedad ambiental es fundamental en sistemas de evaluación de calidad del aire, ya que estas variables influyen directamente en la formación, dispersión y medición de concentraciones de contaminantes como el dióxido de nitrógeno (NO₂) y el ozono (O₃). En la tabla 11 se comparan y evalúan los sensores de temperatura y humedad comerciales disponibles para su implementación.

Tabla 11

Selección de sensor de temperatura y humedad.

Hardware	Requerimientos						Valoración
	StSR4	SrSH3	SrSH4	SrSH9	SrSH10	SrSH11	
SHT1x Temperature & Humidity Sensor	1	1	1	1	1	0	5
Gravity: I2C BME280	1	1	1	1	1	1	6

Environmental

Sensor

SHT40 Temperature & Humidity Sensor	1	0	1	1	1	1	5
--	---	---	---	---	---	---	---

Con la selección realiza en la tabla 11, el sensor que mejor se adapta a los requerimientos es Gravity: I2C BME280 Environmental (Ver figura 11). El sensor ofrece una alta precisión, múltiples funciones y un formato compacto, siendo muy estable en comparación con sensores similares y permite la facilidad de conexiones mediante interfaces SPI e I2C.

Figura 11

Gravity: I2C BME280 Environmental Sensor



Nota. (DFRobot, 2025)

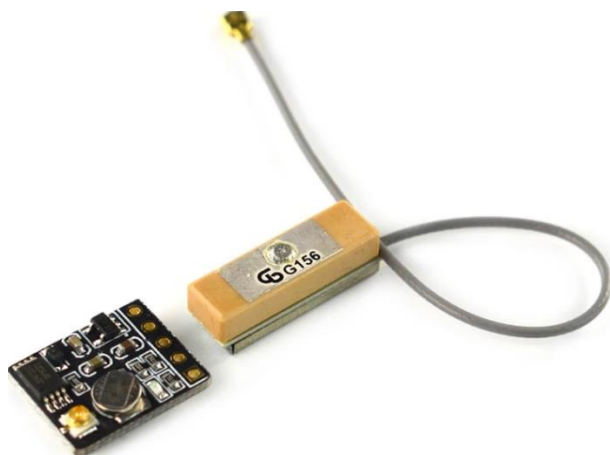
- **Sensor GPS.**

El sensor de GPS permite geolocalizar con precisión el punto exacto donde se realizan las mediciones de calidad del aire. Esta funcionalidad es crucial para mapear la distribución espacial de concentraciones atmosféricas contaminantes. En la tabla 12 se comparan y evalúan los sensores GPS disponibles en el mercado para dispositivos en exteriores.

Tabla 12*Selección de sensor de GPS.*

Hardware	Requerimientos						Valoración
	StSR3	SrSH3	SrSH4	SrSH9	SrSH10	SrSH11	
Módulo GPS Ublox Neo 8M M8N	1	0	1	1	1	1	5
Módulo GPS Ublox Neo 7M 0-001	1	0	1	1	1	0	4
GPS + BDS BeiDou Dual Module	1	1	1	1	1	1	6

Con base en los resultados de la Tabla 12, se seleccionó el sensor GPS + BDS BeiDou Dual Module (véase la Figura 12). Destaca por el rastreo simultáneo de 32 canales (GPS/BDS) y su posicionamiento en tiempo real. Su precisión de 3 metros iguala a los dispositivos comerciales, mientras que su bajo consumo y tamaño reducido favorecen su integración en equipos móviles

Figura 12*GPS + BDS BeiDou Dual Module**Nota.* (DFRobot, 2025)

3.4.2. Selección de Software

A continuación, se presenta la elección del software para el diseño de interfaz web y lenguaje de programación para la programación de los sensores.

3.4.2.1. Selección de framework de desarrollo de la interfaz web.

En esta sección se lleva a cabo la evaluación comparativa del framework seleccionado (Tabla 13), cuyo propósito es sustentar la elección de la herramienta tecnológica para el desarrollo de la plataforma web. Diseñada para mostrar datos en tiempo real, ubicación e historial, esta interfaz requiere un framework de alto rendimiento. Las claves para su selección son la interoperabilidad y la capacidad de integración con tecnologías complementarias.

Tabla 13

Selección de framework de desarrollo de interfaz web.

Framework	Requerimientos					Valoración
	SyRS7	SrRS8	SrRS9	SrSH14	SrSH15	
Flutter web	1	1	1	1	1	5
Angular	1	1	1	1	0	4

El framework que satisface de manera óptima los requerimientos del proyecto es Flutter Web. Si bien Angular es una opción robusta y ampliamente utilizada en el desarrollo de aplicaciones web, su pronunciada curva de aprendizaje representa una barrera significativa para su adopción eficiente. En contraste, Flutter Web proporciona un entorno más accesible y versátil, permitiendo el desarrollo de interfaces web con un alto nivel de personalización y eficiencia, sin comprometer el rendimiento ni la calidad de la experiencia de usuario.

3.4.2.2. Selección de lenguaje de programación.

Para llevar a cabo el procesamiento de datos y la programación de los sensores, es fundamental determinar el lenguaje de programación más adecuado. En la Tabla 14 se presenta la selección del lenguaje utilizado, basada en criterios tomados en los stakeholders.

Tabla 14

Selección de lenguaje de programación.

Lenguaje de programación	Requerimientos					Valoración
	SrSH3	SrSH4	SrSH14	SrSH15	SrSH16	
Matlab	1	0	0	0	1	2
Python	1	1	1	1	1	5
Julia	1	0	1	0	1	3

En base a la valoración realizada se empleará Python, en conjunto con el entorno de distribución de código abierto Anaconda. Esta elección es especialmente adecuada debido a la amplia adopción de Anaconda en los campos de la ciencia de datos y el aprendizaje automático, así como por su capacidad para gestionar paquetes y dependencias de manera eficiente, proporcionando un entorno de desarrollo estable y robusto (Anaconda, 2024).

Se prioriza el uso de Python sobre MATLAB debido a su flexibilidad y la disponibilidad de bibliotecas avanzadas (TensorFlow, Keras, Scikit-Learn). Esto evita los costos de licencia asociados a MATLAB y optimiza el flujo de trabajo en la creación de modelos predictivos y análisis de datos.

3.5. Formación de dataset.

Para la formación del dataset destinado al entrenamiento de una red neuronal orientada a la predicción de niveles de contaminación del aire, se utilizó la plataforma World's Air Pollution: Real-time Air Quality Index (WAQI). Esta fuente proporciona datos

en tiempo real sobre diversos contaminantes atmosféricos y condiciones meteorológicas, recopilados a través de una red global de más de 10,000 estaciones de monitoreo. La elección de esta plataforma se fundamenta en su amplia cobertura geográfica y en la disponibilidad de datos históricos y actuales, esenciales para el análisis temporal requerido en modelos de aprendizaje automático.

La recolección de datos se realizó mediante la API JSON proporcionada por WAQI, la cual permite el acceso programático a información detallada sobre contaminantes como el dióxido de nitrógeno (NO₂), ozono (O₃), así como variables meteorológicas como temperatura y humedad relativa. La API ofrece funcionalidades adicionales, incluyendo consultas geolocalizadas, acceso a datos de estaciones específicas y previsiones de calidad del aire, lo que facilita la obtención de un conjunto de datos robusto y representativo para el área de estudio

3.5.1. Categorización de variables ambientales según niveles de riesgo.

Con el objetivo de facilitar la clasificación supervisada de los niveles de calidad del aire mediante redes neuronales, se definieron rangos discretos para cada una de las variables ambientales analizadas: dióxido de nitrógeno (NO₂), ozono (O₃), temperatura y humedad relativa. Esta discretización permitió asignar una etiqueta de clase (estimates) a cada registro del dataset, categorizando los niveles de afectación a la salud en: Limpio, Elevado e Insalubre.

Los umbrales fueron determinados de manera empírica a partir de la distribución observada en los datos, tomando como base los criterios referenciales de organismos internacionales como la OMS y la EPA, así como los estándares de la plataforma AQI.

A continuación, se presenta la tabla 15 con los rangos establecidos para cada categoría.

Tabla 15*Rangos de niveles de contaminación.*

Variable	Limpio	Elevado	Insalubre
NO2(ppm)	0.00 – 0.10	0.11 – 0.16	>0.16
O3(ppb)	0 – 80	81 – 100	>100
Temperatura (°C)	15 – 26	26 – 30	>30
Humedad (%)	55 – 70	70 – 80	>80

3.6. Diseño del sistema

En esta sección se desarrolla la siguiente etapa de la fase de verificación y simulación de la metodología *Bottom-Up*, comenzando con el diseño del sistema. Durante esta fase, se ejecutan los procesos fundamentales para la concepción y desarrollo de sus funcionalidades. En este contexto, se define el modelo de arquitectura IoT del sistema, acompañado de los diagramas de bloques y de flujo, los cuales representan gráficamente el funcionamiento del sistema. Asimismo, se elabora el diagrama de conexión de los componentes de hardware, proporcionando una representación detallada de la interconexión de los dispositivos electrónicos, como sensores y placa de procesamiento.

3.6.1. Arquitectura IoT del Sistema

El modelo de arquitectura IoT del sistema, fundamentado en el paradigma de Internet de las Cosas, se ilustra en la Figura 13. Este modelo está compuesto por cuatro capas principales: capa de dispositivos, capa de red, capa de servicio y capa de aplicación, cada una con una función específica y esencial para garantizar un sistema integral y operativo. A continuación, se describen en detalle las funcionalidades asociadas a cada capa.

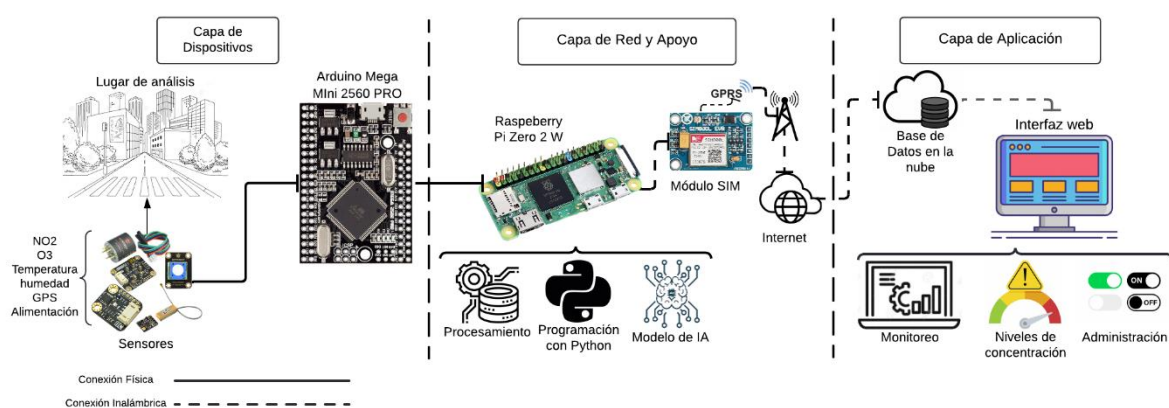
- **Capa de Dispositivos:** Agrupa los dispositivos físicos de detección (sensores de NO2, O3, clima y GPS) y el sistema de energía. Su función es capturar las

variables del entorno y asegurar el suministro eléctrico necesario para la operación de todos los componentes.

- Capa de Red y apoyo:** Esta capa se divide en dos funciones principales. La capa de red se encarga de la transmisión de los datos obtenidos por los sensores mediante una conexión inalámbrica con apoyo de un módulo SIM que permita una red de datos móviles e internet, estableciendo la comunicación con la capa de aplicación en la nube. Por otro lado, la capa de apoyo se encarga del procesamiento inicial de los datos y de la ejecución del modelo de red neuronal antes de su envío, optimizando la información para su análisis en niveles superiores del sistema
- Capa de Aplicación:** Esta capa la interfaz web recolecta del almacenamiento en la nube los datos recopilados y procesados, permitiendo su gestión y acceso. En esta interfaz, se visualizan los niveles de concentración de los gases, además de ofrecer herramientas para el monitoreo y la administración del sistema.

Figura 13

Arquitectura IoT del Sistema.



3.6.2. Diagrama de Bloques General del Sistema.

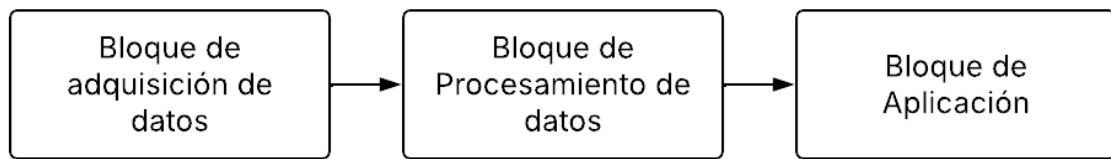
La Figura 14 ilustra el diagrama de bloques general del sistema, en el cual se integran los componentes para representar la funcionalidad del sistema de monitoreo de la calidad del

aire. Este diagrama se estructura en cuatro bloques interconectados, cuya interacción permite el cumplimiento del objetivo funcional del sistema. A continuación, se presentan los detalles específicos de cada uno de estos bloques.

- **Bloque de adquisición de datos:** Este bloque garantiza primero el suministro energético del sistema. Su función central es la recolección de métricas ambientales (NO₂, O₃, clima y GPS), las cuales son capturadas por un Arduino Mega Mini. Posteriormente, la información se transmite a la Raspberry Pi Zero 2W para el procesamiento de datos y la generación de modelos predictivos.
- **Bloque de procesamiento:** El procesamiento recae en la Raspberry Pi Zero 2W, que emplea Python para analizar datos y ejecutar una red neuronal preentrenada destinada a estimar contaminantes. Una vez procesados, los resultados se transmiten a una base de datos en la nube vía Wi-Fi mediante peticiones HTTP.
- **Bloque de aplicación:** Este bloque comprende la interfaz web desarrollada mediante el framework Flutter, la cual permite la visualización y gestión de la información almacenada en la base de datos. La aplicación web está diseñada para presentar en tiempo real los datos del sistema de monitoreo de calidad del aire, proporcionando una interfaz intuitiva para la supervisión de las variables ambientales. Además, incorpora un módulo de administración que facilita la gestión, configuración de la plataforma y el sistema, asegurando su operatividad o cambios posteriores.

Figura 14

Diagrama de Bloques General del Sistema.



3.6.3. Diagramas de Flujo.

En esta sección se describen los diagramas de flujo del procesamiento de los datos, entrenamiento de la red y adquisición del modelo respectivamente.

3.6.3.1 Flujograma de procesamiento de datos.

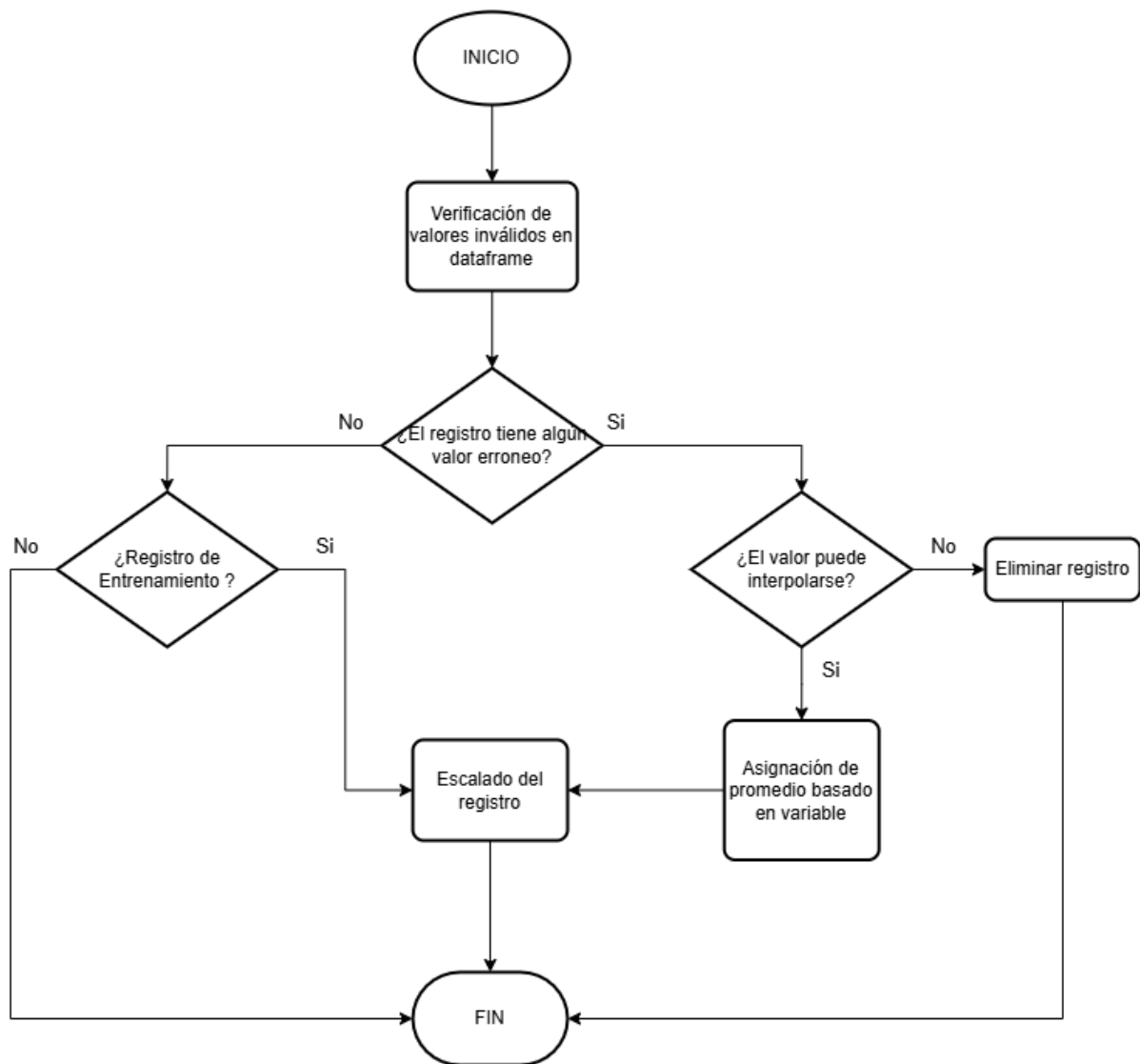
El diagrama de flujo de la Figura 15 inicia el procesamiento necesario que requieren los datos una vez han sido recolectados por los sensores. Este procedimiento tiene como objetivo garantizar la calidad de los datos antes de ser utilizados para la predicción o de entrenamiento de modelos predictivos.

La fase comienza verificando la integridad de los datos y filtrando valores erróneos. Para los registros de entrenamiento válidos, es crucial aplicar el escalado, asegurando así la normalización necesaria para el correcto funcionamiento del modelo.

Se prioriza la corrección de datos mediante interpolación guiada por la variable más correlacionada. Si esto no es posible, el registro se descarta para mantener la integridad del set. Esta lógica de depuración actúa como un módulo estándar en los esquemas posteriores.

Figura 15

Diagrama de Flujo de procesamiento de datos.



3.6.3.2. Flujograma de entrenamiento de la red.

El entrenamiento de la red neuronal se lleva a cabo con Jupyter Notebook de Anaconda, similar a otras herramientas de desarrollo en ciencia de datos, pero utilizando recursos computacionales locales que permiten mayor control sobre la ejecución del código y la administración del entorno. En la Figura 16, el proceso de entrenamiento del modelo se divide en varios subprocesos organizados de manera secuencial y lógica para asegurar una correcta preparación de los datos y una adecuada configuración del modelo.

El flujo inicia con la carga del dataset en Jupyter, lo cual marca el punto de partida para la manipulación de los datos. Posteriormente, se realiza el procesamiento de los datos,

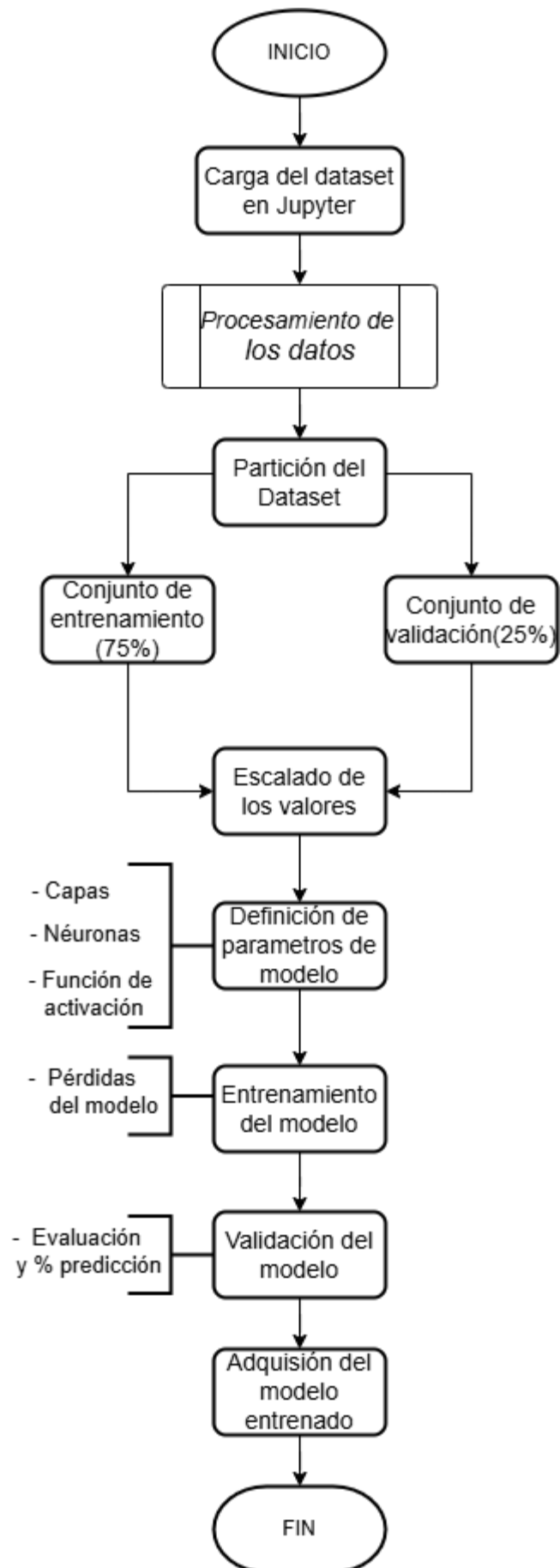
que hace referencia al diagrama de flujo presentando en la figura 15, donde se efectúan tareas preliminares.

Una vez procesado, el dataset es particionado en dos subconjuntos: un conjunto de entrenamiento (75%), que se utiliza para ajustar los parámetros del modelo, y un conjunto de validación (25%), destinado a evaluar su desempeño durante el proceso de entrenamiento, porcentajes acertados para la dimensión del dataset (Géron, 2019). El proceso requiere escalar las variables para evitar que alguna domine el aprendizaje. Luego se definen la estructura y funciones del modelo (capas, activación, pérdida) y se ejecuta el entrenamiento, permitiendo que el sistema optimice sus pesos según el error detectado.

Seguido del entrenamiento, se ejecuta la validación del modelo, en la cual se evalúa su desempeño utilizando el conjunto de validación. Esta fase incluye el cálculo del porcentaje de precisión y otros indicadores de evaluación. Finalmente, se realiza la adquisición del modelo entrenado, lo cual implica guardar la configuración y los pesos aprendidos para su posterior uso en tareas de inferencia.

Figura 16

Diagrama de Flujo de entrenamiento de la red Neuronal.



3.6.3.3. Diagrama de Flujo del sistema.

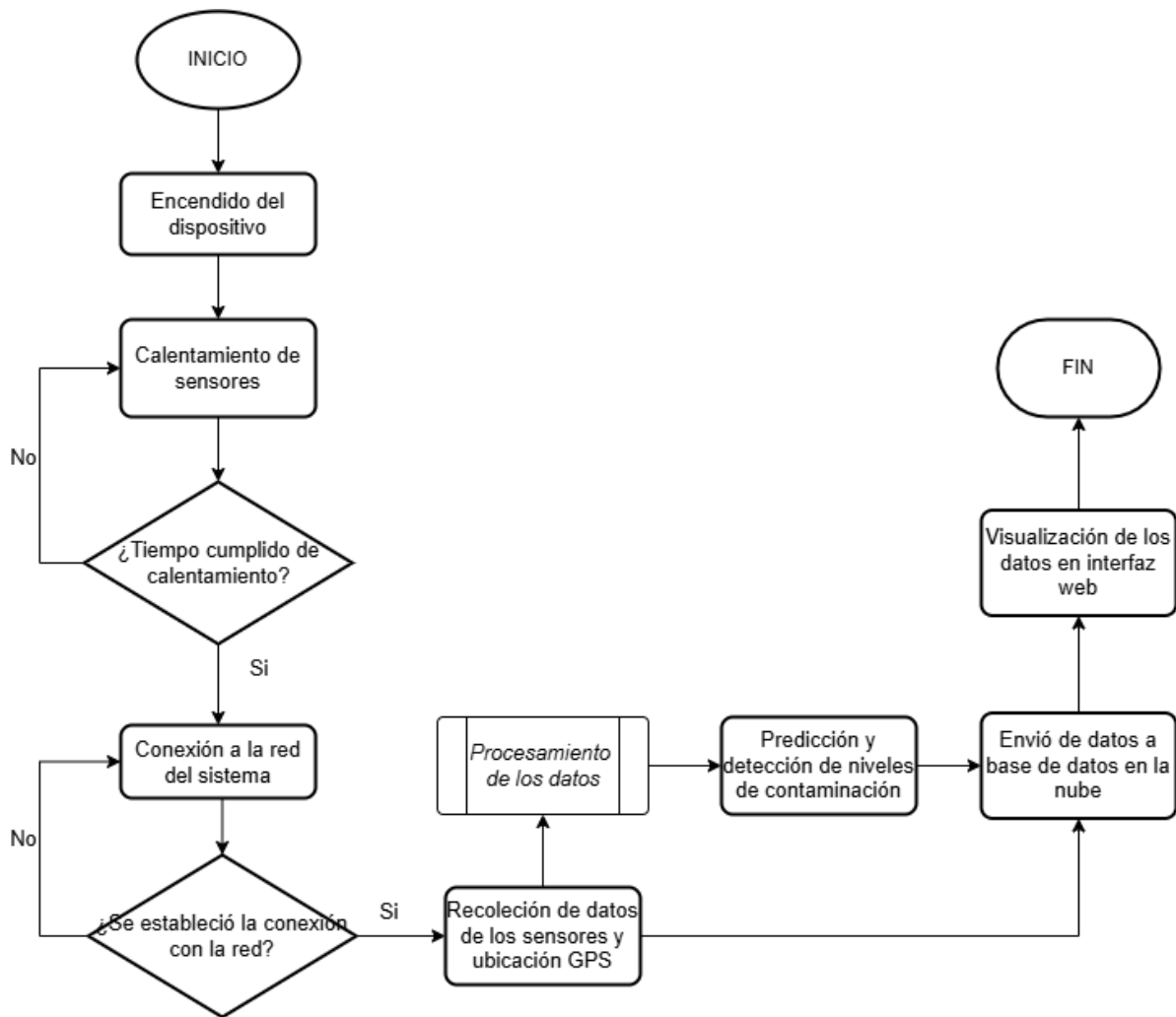
El funcionamiento general del sistema de monitoreo de calidad del aire se basa en el diagrama de flujo representado en la Figura 17. Dado que el sistema opera como un único nodo, se contempla su funcionamiento integral, comenzando con el encendido y el estado de espera (*standby*), el cual se mantiene hasta que los sensores completen su proceso de calentamiento, requerido debido a sus propiedades electroquímicas.

Una vez transcurrido el tiempo estimado de calentamiento, el sistema procede a establecer la conexión con la red. En caso de que la conexión no se logre, el sistema permanecerá en este estado hasta conseguirla. Una vez establecida la conexión, se inicia la recolección de datos provenientes de los sensores, seguida de su procesamiento y formateo que incluye tareas de limpieza y escalamiento haciendo referencia al diagrama presentado en la figura 15, necesarias para la predicción y detección de niveles de contaminación mediante el modelo de red neuronal desarrollado para su posterior envío a la base de datos en la nube.

Finalmente, cuando los datos se encuentran en la base de datos almacena la información y la transmite a la interfaz web, donde los datos son publicados para su visualización en tiempo real.

Figura 17

Diagrama de Flujo del sistema.



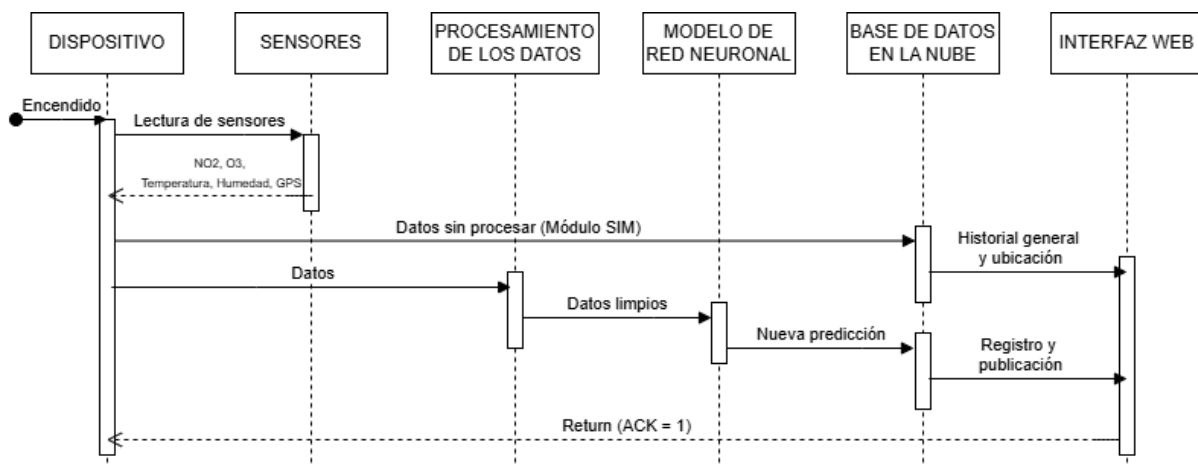
En la interacción de los módulos del sistema se refleja en el siguiente diagrama de secuencia (figura 18), donde se muestra el flujo completo de datos que realiza el dispositivo. El proceso se inicia con el encendido del dispositivo, lo cual activa la lectura de sensores integrados en el sistema. Estos sensores capturan parámetros relevantes como niveles de dióxido de nitrógeno (NO₂), ozono (O₃), temperatura, humedad y la ubicación geográfica mediante GPS.

Una vez obtenidos estos datos, se transmiten como datos sin procesar al módulo de procesamiento de datos, donde se lleva a cabo una limpieza de datos para eliminar valores erróneos o inconsistentes. Esta etapa garantiza la integridad y confiabilidad de la información antes de su análisis posterior.

Posteriormente, los datos limpios se envían al modelo de red neuronal, que ejecuta una nueva predicción basada en patrones aprendidos previamente. Su objetivo es inferir condiciones ambientales y registrar los resultados en la nube, manteniendo actualizado el historial y la ubicación. Los datos se presentan en una plataforma web y el proceso culmina con una señal de confirmación (ACK); un valor de 1 valida el envío y autoriza la continuidad de la transmisión. En cambio, un valor de 0 señala un fallo en la transmisión, por lo que el sistema mantiene activo el proceso de reenvío.

Figura 18

Diagrama de secuencia del sistema.



3.6.4. Comunicación del sistema.

La comunicación del sistema se establece mediante el uso de redes móviles, específicamente mediante el protocolo GPRS, lo que permite la transmisión de datos desde el nodo de monitoreo ambiental hacia un servidor remoto para su almacenamiento, análisis y visualización posterior. Esta vía de comunicación se selecciona debido que permite la independencia del sistema, y a su amplia cobertura geográfica que ofrece, características especialmente valiosas en entornos exteriores urbanos intermedios.

Los datos se envían periódicamente usando HTTP, un protocolo eficaz para conexiones como GPRS. Así, los niveles de contaminantes y variables del entorno se transmiten en tiempo real o cuasi-real para su monitoreo en la web.

Se contempla el uso de la conexión Wi-Fi propia de la placa como mecanismo de contingencia. Esta opción permite mantener la transmisión de datos si falla la red móvil y facilita tareas de configuración remota, asegurando la continuidad del servicio frente a interrupciones.

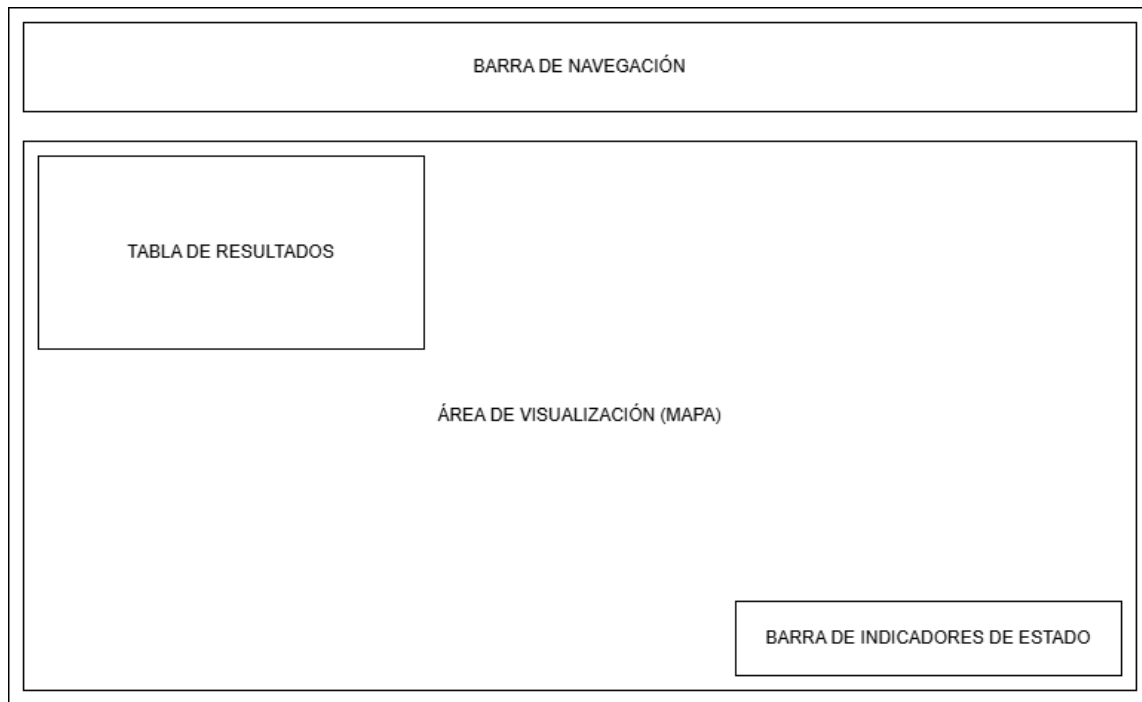
3.6.5. Diseño general de interfaz Web.

El diseño de la interfaz web se ha estructurado cuidadosamente para facilitar la visualización, navegación y comprensión de los datos recolectados por el sistema de monitoreo ambiental. A través de una organización modular y jerárquica de los elementos gráficos, se garantiza una experiencia de usuario clara, eficiente y funcional.

En la Figura 19 se presenta la estructura correspondiente a la pantalla principal de la interfaz web, la cual se muestra de manera predeterminada al momento de acceder al sistema. Orientado a la visualización geoespacial, el sistema cuenta con una barra de navegación permanente. Su cuerpo central combina una tabla de datos, ideal para revisiones numéricas precisas, con un mapa que proyecta gráficamente la ubicación y distribución de las variables monitoreadas. Como elemento complementario, se integra una barra de indicadores de estado, ubicada estratégicamente en la parte inferior derecha. Esta sección resume las condiciones ambientales actuales mediante el uso de elementos gráficos visuales y fácilmente interpretables, brindando una visión rápida del estado del entorno monitorizado.

Figura 19

Diseño general de pantalla predeterminada



Por su parte, la Figura 20 muestra una segunda pantalla, cuya estructura responde a un enfoque de seguimiento temporal y sintético de los datos. Bajo la misma barra de navegación fija, la pantalla organiza la información en dos áreas: un historial central para el seguimiento temporal de los datos y un panel inferior de estadísticas que resume el comportamiento general del sistema.

Figura 20

Diseño general de pantalla de resultados.



3.6.6. Plataforma de despliegue de interfaz web.

Se emplea Netlify como entorno de producción debido a su soporte robusto para frameworks como Flutter Web. Sus funcionalidades de despliegue automático desde Git y distribución mediante CDN aseguran la disponibilidad y eficiencia necesarias para el monitoreo ambiental remoto (Netlify, 2025).

Tras compilar el proyecto en archivos HTML, CSS y JavaScript, Netlify gestiona la publicación automática desde el repositorio. Esto habilita el acceso remoto inmediato, permitiendo a los interesados monitorear las estimaciones de contaminación en tiempo real desde cualquier lugar.

3.6.7. Diagrama esquemático de conexiones del sistema.

El diagrama esquemático de conexiones del sistema, representado en la Figura 21, ilustra todas las interconexiones necesarias para su correcto funcionamiento. La alimentación de la placa Raspberry Pi Zero 2W, se realiza a través del puerto Micro-USB Power In, diseñado específicamente para este propósito y que requiere una fuente de 5V y 2A. No

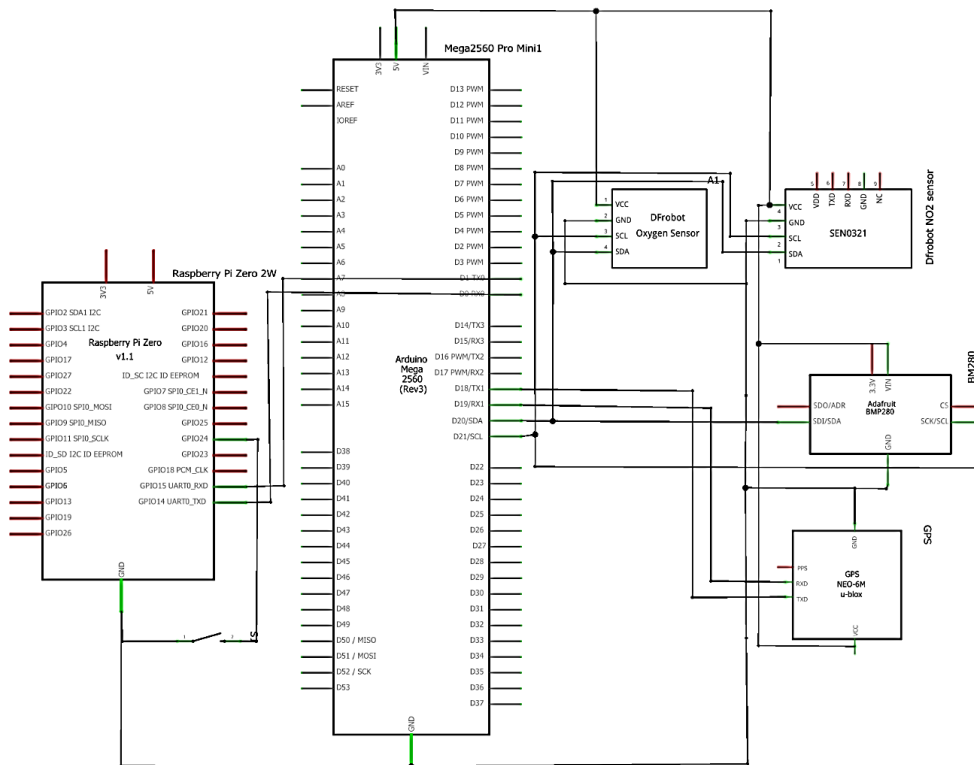
obstante, la placa también puede ser alimentada mediante los pines de conexión, siempre que se utilice una fuente de energía estable, sin picos de tensión que puedan comprometer su operatividad.

El Arduino Mega Mini realiza la lectura y alimentación de los sensores de NO₂ y O₃ que operan con una alimentación de 5V y disponen de dos opciones de comunicación. En este caso, se emplea la interfaz I2C por defecto, ya que permite la conexión simultánea de múltiples dispositivos en el mismo bus, asignando a cada sensor una dirección única. Para el sensor BME280, encargado de medir temperatura y humedad, se emplea una comunicación de I2C en la misma conexión simultánea, y su alimentación es de 5v.

En el caso del módulo GPS se alimenta con 5v, utiliza una comunicación UART por defecto. Por último, la comunicación entre el Arduino Mega Mini y Raspberry Pi Zero 2W se realiza por comunicación serial. Se requiere un proceso de apagado por software para el caso de la Raspberry Pi Zero 2W para garantizar la integridad del sistema y la placa, se incorpora un botón de apagado que utiliza la función de pull-up propio de la placa. Para ejecutar el apagado completo del sistema, mantendrá presionado el botón durante unos segundos.

Figura 21

Diagrama esquemático de conexiones del sistema.



3.6.8. Consumo energético del sistema.

En esta sección se lleva a cabo un análisis detallado del consumo energético total del sistema, con el propósito de determinar la cantidad de energía necesaria para asegurar su correcto funcionamiento y estimar su autonomía operativa. La Tabla 15 resume los componentes que integran el sistema, junto con sus respectivas especificaciones de consumo energético.

Tabla 16

Consumo energético de los componentes.

Componente	Voltaje (VDC)	Corriente (mA)
Raspberry Pi Zero 2W	5 V	120 mA
Arduino Mega Mini 250 Pro	5 V	80 mA
Gravity: Electrochemical	5 V	5 mA
Nitrogen Dioxide Sensor		

Gravity: Electrochemical	5 V	5 mA
Ozone Sensor		
Gravity: I2C BME280	5 V	2 mA
Environmental Sensor		
GPS + BDS BeiDou Dual	5 V	25 mA
Module		
Modulo SIM800L	5 V	500 mA
Total	5 V	737 mA

A partir de los datos presentados en la Tabla 16, se establece que, para garantizar el funcionamiento continuo del sistema durante un periodo de 8 horas necesario para análisis prolongados, es preciso calcular la capacidad requerida de la batería en amperios-hora (Ah). Este cálculo se realiza mediante la Ecuación 1:

$$\text{Capacidad (Ah)} = \text{Corriente total (A)} \times \text{Tiempo (h)}$$

$$\text{Capacidad (Ah)} = 0.737 \text{ A} \times 8 \text{ h} \approx 5896 \text{ mAh}$$

Esto implica que, considerando una tensión de operación de 5 V, se requiere una batería capaz de suministrar aproximadamente 5896 mAh para mantener el sistema activo durante el tiempo especificado.

3.6.9. Diseño del Case del sistema.

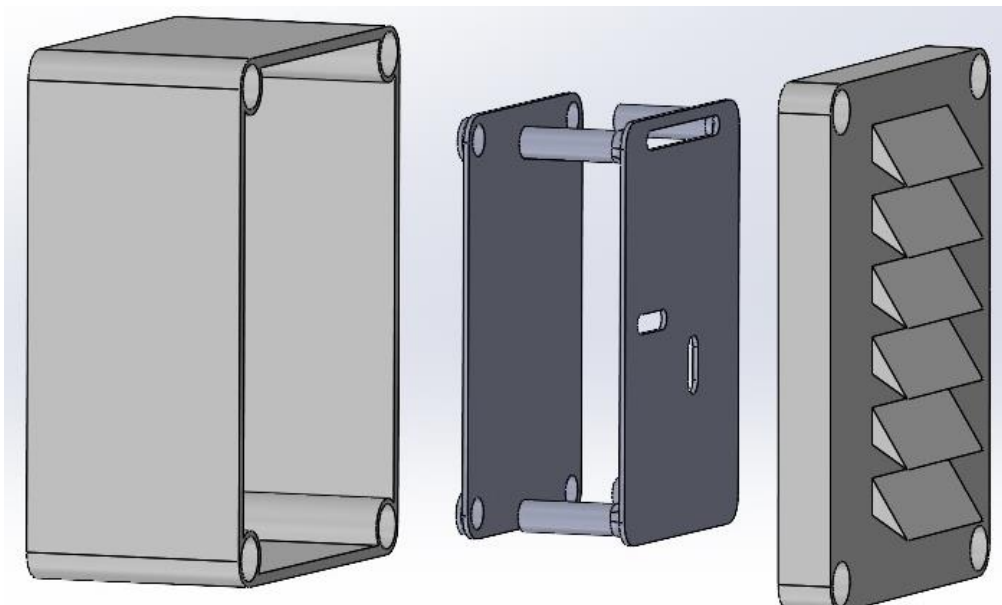
Para garantizar la protección y el adecuado funcionamiento del sistema IoT en condiciones ambientales reales durante las pruebas de campo, se optó por el uso de una caja de protección con grado de estanqueidad IP55, modelo CD*18148. Este tipo de caja ofrece resistencia frente al ingreso de polvo y chorros de agua, lo cual la hace adecuada para entornos exteriores y urbanos donde se desarrollarán las jornadas de monitoreo.

Para asegurar la correcta disposición y fijación de los componentes electrónicos del sistema, se diseñaron soportes personalizados en 3D. Como se muestra en la Figura 22, estos modelos permiten organizar los elementos internos de forma eficiente, facilitan el mantenimiento y mejoran la ventilación pasiva.

Además, se realizó una modificación en la tapa exterior de la caja para permitir la lectura adecuada de las variables ambientales sin comprometer la protección del sistema.

Figura 22

Diseño de case del sistema.



3.7. Implementación del sistema.

Esta sección detalla la implementación técnica tras las fases de requerimientos y diseño. El proceso se divide en tres etapas fundamentales: el entrenamiento del modelo de redes neuronales, la integración del hardware para la medición de NO₂ y O₃, y el desarrollo de la interfaz web para la visualización y monitoreo de los datos.

3.7.1. Construcción de modelo de red neuronal.

Para el entrenamiento de la red neuronal, fue indispensable etiquetar cada registro con una referencia confiable del nivel de contaminación. Ante la carencia de estaciones de monitoreo locales, se seleccionó la plataforma World Air Quality Index (WAQI) como fuente principal. Esta herramienta proporciona estimaciones del Índice de Calidad del Aire (ICA) mediante datos satelitales y sensores comunitarios, permitiendo asignar una categoría cualitativa a cada muestra antes de su almacenamiento. Esta clasificación sirvió como variable dependiente (output) para el entrenamiento supervisado de la red neuronal.

Las etiquetas empleadas fueron definidas empíricamente con base en los rangos del ICA y se agruparon en tres niveles de calidad del aire:

- Limpio o Aire salubre
- Elevado o Calidad del aire subóptima
- Insalubre

Esta categorización permitió formular el problema como una tarea de clasificación multiclase, facilitando así el diseño del modelo predictivo. Con este enfoque, se construyó un conjunto de datos compuesto por 3270 muestras representativas, cada una etiquetada de acuerdo con su nivel estimado de contaminación, sirviendo como base para el entrenamiento y validación del modelo neuronal.

3.7.1.1. Preprocesamiento

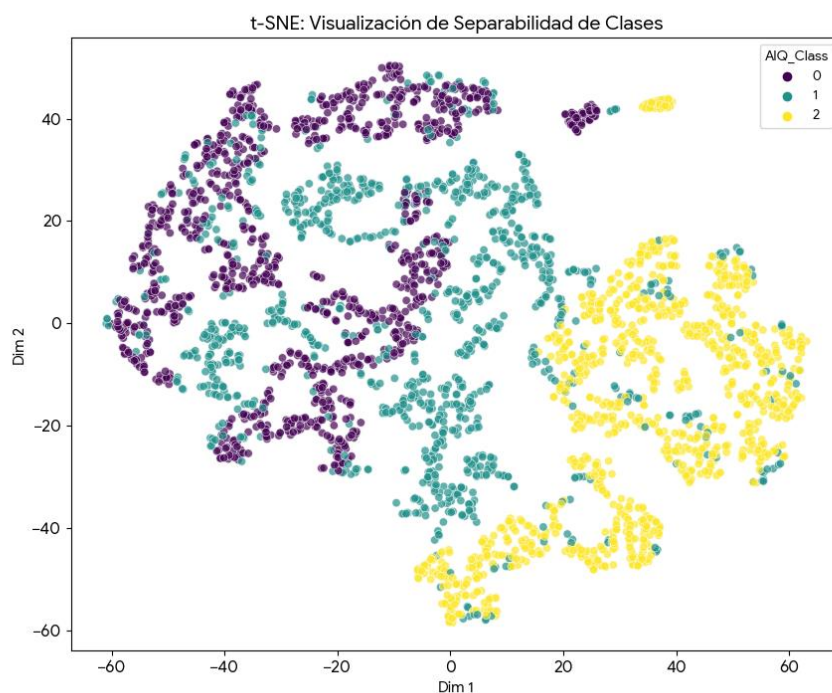
El preprocesamiento constituye una de las etapas más críticas en el diseño de cualquier modelo de aprendizaje automático, ya que garantiza que los datos se encuentren en condiciones óptimas para su análisis y procesamiento por parte del algoritmo. En este proyecto, los datos provienen directamente de sensores ambientales que registran concentraciones de gases contaminantes como NO₂ y O₃. Debido a la estabilidad y confiabilidad del sistema de adquisición, no fue necesario aplicar técnicas de limpieza ni

interpolación, ya que no se detectó la presencia de valores nulos ni inconsistencias en los registros.

En lugar de aplicar procesos de imputación o eliminación de datos, se optó por realizar una exploración visual de las características del conjunto de datos mediante técnicas de análisis exploratorio avanzadas. Este enfoque permitió comprender la estructura interna de las variables y evaluar la separabilidad entre las clases de la variable objetivo, correspondiente a los niveles de calidad del aire: Limpio, Elevado e Insalubre como se muestra en la siguiente figura.

Figura 23

Visualización de Separabilidad de Clases.



La Figura 23 emplea la técnica t-SNE para visualizar patrones complejos, revelando una clara diferenciación de la clase Insalubre, influenciada por los niveles de O₃. Por el contrario, las categorías Limpio y Elevado muestran solapamientos coherentes con sus rangos intermedios. Esta estructura topológica confirma la naturaleza no lineal del problema,

justificando técnicamente la elección de una red neuronal frente a algoritmos lineales para abordar la clasificación.

3.7.1.2. Escalado

El escalado de las variables de entrada es una etapa crítica en el entrenamiento de redes neuronales, debido a que estos modelos son sensibles a las diferencias de magnitud entre características. En este trabajo se empleó la técnica de estandarización mediante `StandardScaler`. Este método transforma los datos centrando su distribución en una media de cero y ajustando su varianza a una desviación estándar de uno, lo cual facilita una convergencia más rápida y estable del algoritmo de optimización, como se muestra en la figura 24.

Figura 24.

Escalado de datos antes de la división.

```
X = df.drop('AIQ_Class', axis=1)
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Esta técnica resulta especialmente adecuada considerando que las variables provienen de sensores que registran contaminantes en unidades heterogéneas, como partes por billón (PPB) y partes por millón (PPM). El uso de esta normalización contribuye a una convergencia más estable y eficiente durante el proceso de entrenamiento de la red.

3.7.1.3. División de datos de entrenamiento y prueba.

Una vez completado el análisis preliminar y verificado el balance de clases en la variable objetivo, se procedió a la partición del conjunto de datos en dos subconjuntos: uno destinado al entrenamiento del modelo y otro para su evaluación posterior. Este procedimiento se implementó mediante la función `train_test_split` de la biblioteca *scikit-learn*, estableciendo una proporción de 80% para entrenamiento y 20% para prueba, tal como se ilustra en la Figura 25. Se fijó una semilla aleatoria (`random_state=42`) para garantizar la

reproducibilidad de los resultados. Esta división es esencial para evitar el sobreajuste y asegurar la capacidad de generalización del modelo frente a datos no observados.

Figura 25

División de datos de entrenamiento.

```
y = df['AIQ_Class']
y_encoded = to_categorical(y)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)
```

Esta división es esencial para evitar que el modelo memorice los datos de entrada y asegure su capacidad de generalización frente a datos no vistos. En particular, la codificación de la variable objetivo se realizó mediante la técnica *one-hot encoding*, utilizando *to_categorical*, lo cual es apropiado para tareas de clasificación multiclase

3.7.1.4. Creación del modelo.

Para el desarrollo del modelo predictivo, se implementó una red neuronal artificial de tipo Perceptrón Multicapa (MLP) utilizando la arquitectura secuencial de la biblioteca Keras (*TensorFlow*). Esta estructura se diseñó para procesar un vector de entrada de cuatro dimensiones, correspondiente a las variables predictoras previamente estandarizadas.

Como se muestra en la figura 26 la topología de la red consta de dos capas ocultas densamente conectadas (*fully connected*). La primera capa oculta posee 64 neuronas, mientras que la segunda cuenta con 32 neuronas. Para ambas capas intermedias se seleccionó la función de activación ReLU (*Rectified Linear Unit*), la cual es ampliamente utilizada en aprendizaje profundo debido a su eficiencia computacional y su capacidad para introducir no linealidad en el modelo, mitigando problemas como el desvanecimiento del gradiente. El modelo incluye capas de Dropout al 20% para evitar la saturación y mejorar la predicción en datos nuevos. Finalmente, la capa de salida consta de tres neuronas que se alinean con las clases a predecir. En esta última capa se empleó la función de activación *Softmax*, la cual es idónea para problemas de clasificación multiclase, ya que transforma las salidas numéricas de

la red en una distribución de probabilidad normalizada, donde la suma de los valores resultantes es igual a uno.

Figura 26

Arquitectura general del modelo.

```
model = Sequential([
    Dense(64, input_shape=(4,)), activation='relu'),
    Dropout(0.2),
    Dense(32, activation='relu'),
    Dropout(0.2),
    Dense(3, activation='softmax')
])
```

3.7.1.5. Compilación y entrenamiento del modelo.

El modelo fue compilado utilizando el optimizador Adam, ampliamente utilizado por su capacidad adaptativa y eficiencia computacional. La función de pérdida seleccionada fue *categorical_crossentropy*, que es la más adecuada para problemas de clasificación multiclase con etiquetas codificadas en *one-hot*.

Figura 27

Parámetros de compilación del modelo.

```
model.compile(optimizer='adam',
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

Durante el entrenamiento, se implementaron dos técnicas de control:

- *EarlyStopping*: para detener el proceso si no se observan mejoras en la pérdida de validación durante 10 épocas consecutivas.
- *ReduceLRonPlateau*: para disminuir la tasa de aprendizaje si la pérdida de validación se estanca, mejorando la eficiencia en fases avanzadas del entrenamiento.

Figura 28

Funciones de control del entrenamiento.

```
# EarlyStopping: Detiene el entrenamiento si no mejora tras 'patience' épocas
early_stopping = EarlyStopping(monitor='val_loss',patience=10,restore_best_weights=True,verbose=1)

# ReduceLRonPlateau: Reduce la velocidad de aprendizaje si se estanca
reduce_lr = ReduceLRonPlateau(monitor='val_loss',factor=0.5,patience=5,min_lr=1e-6,verbose=1)
```

Para determinar la duración óptima del entrenamiento, se evaluaron configuraciones con límites máximos de 20, 40 y 100 épocas. Se observó que fijar un límite bajo se completaba el aprendizaje sin interrupción, mientras que el límite de 100 épocas permitió al modelo converger adecuadamente, actuando en conjunto con los mecanismos de control de sobreajuste como se muestra en la figura 29.

Aunque se estableció un máximo de 100 épocas, el entrenamiento no requirió completar la totalidad del ciclo. El mecanismo de parada temprana (*Early Stopping*) intervino automáticamente deteniendo el proceso en la época 71, tras detectar que la pérdida en el conjunto de validación no mostraba mejoras significativas durante el periodo de paciencia establecido. Simultáneamente, el *callback ReduceLRonPlateau* ajustó dinámicamente la tasa de aprendizaje para afinar la búsqueda del mínimo global en la función de pérdida.

Como resultado de esta estrategia, se restauraron los pesos del modelo correspondientes a la época 61, punto donde se registró el mejor desempeño de generalización. La evaluación final sobre el conjunto de prueba, utilizando estos pesos optimizados, arrojó una precisión del 97.64%, demostrando un equilibrio robusto entre el aprendizaje de patrones y la capacidad de predicción ante nuevos datos.

Figura 29

Entrenamiento del modelo.

```

history = model.fit(
    X_train, y_train,
    epochs=100,
    batch_size=32,
    validation_split=0.2,
    callbacks=[early_stopping, reduce_lr]
)

```

En la Figura 30 se ilustra el comportamiento del modelo a lo largo de distintas épocas, lo cual permite una evaluación comparativa de la dinámica de aprendizaje, el proceso de convergencia y la capacidad de generalización conforme aumenta el número de iteraciones de entrenamiento. Cada fila presenta un gráfico donde se muestra el historial de pérdida y, el historial de precisión, ambos mostrando el desempeño del modelo tanto sobre el conjunto de entrenamiento como sobre el de validación.

Figura 30

Gráficos del comportamiento del modelo en 100 épocas.

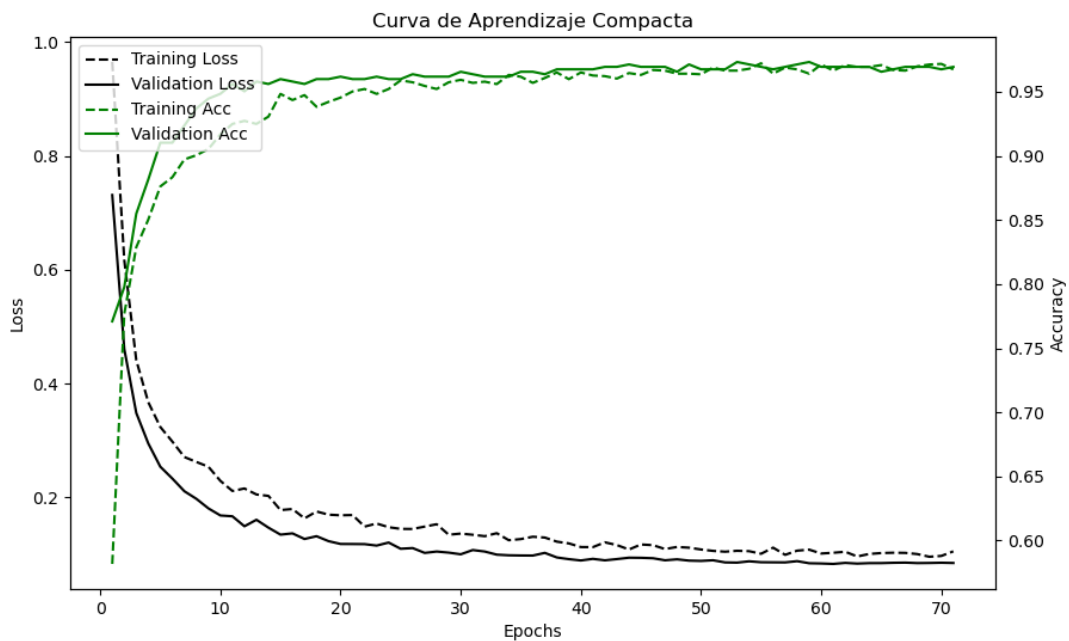


Figura 31

Gráficos del comportamiento del modelo en 40 épocas.

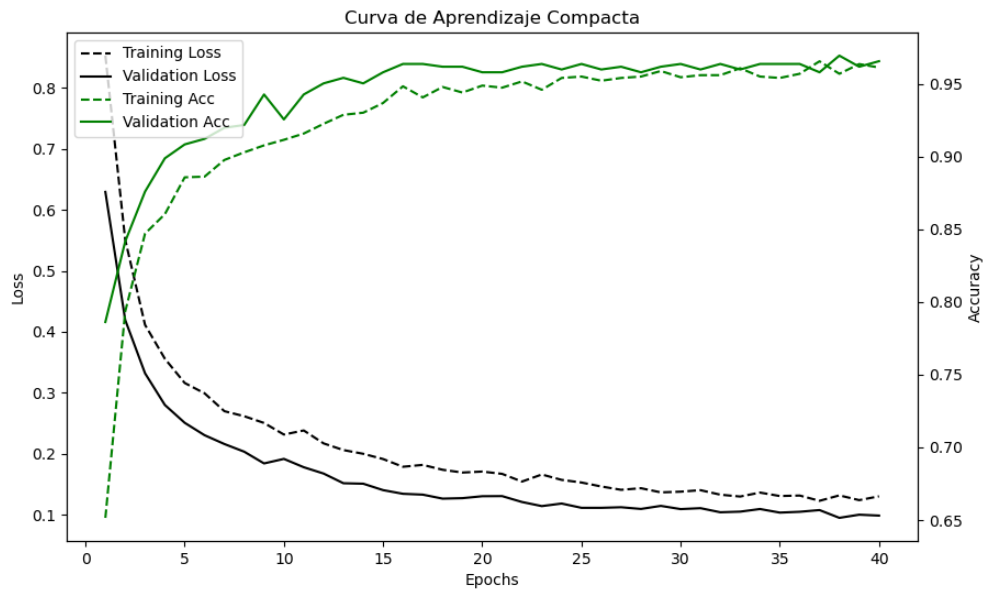
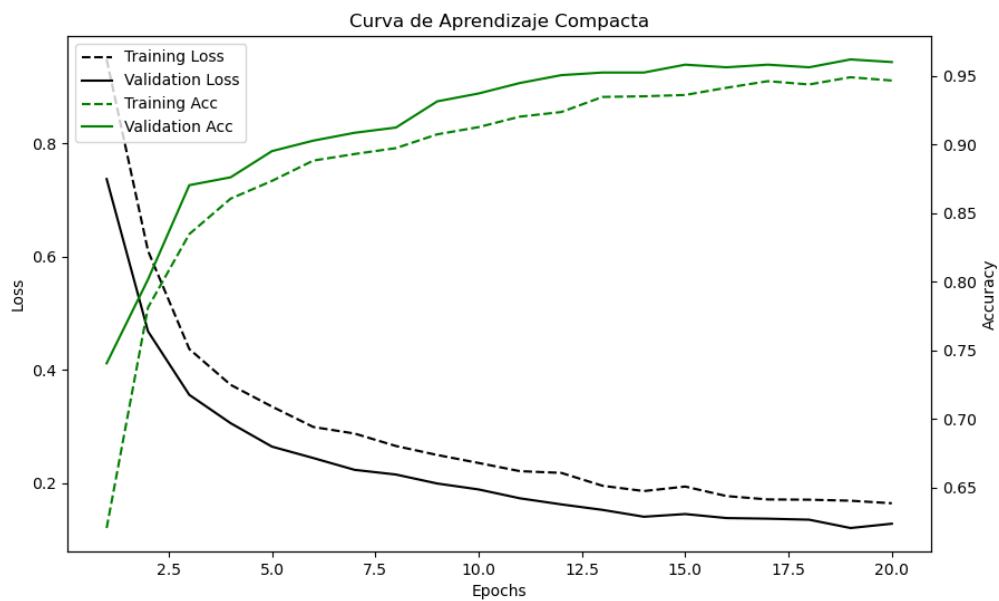


Figura 32

Gráficos del comportamiento del modelo en 20 épocas.



3.7.1.6. Evaluación y rendimiento del modelo.

A lo largo de las diferentes fases de experimentación, se evaluó el comportamiento de la red neuronal bajo distintos horizontes de entrenamiento, limitados a 20, 40 y un ciclo extendido controlado por parada temprana, tal como se ilustra en las Figuras 30, 31 y 32 respectivamente.

El análisis de las curvas de aprendizaje revela una convergencia rápida y estable. En las configuraciones preliminares de 20 y 40 épocas, se observó que el modelo alcanzaba altos niveles de precisión desde las primeras iteraciones, superando el 90% tanto en entrenamiento como en validación. Sin embargo, fue en el ciclo final (Figura 32), donde actuó el mecanismo de Early Stopping deteniendo el proceso en la época 71, donde se logró la mayor estabilidad. En este escenario, la curva de pérdida muestra un descenso sostenido y suave, mientras que la brecha entre la línea de entrenamiento y la de validación se mantiene mínima, lo cual descarta problemas de sobreajuste (overfitting) y confirma una excelente capacidad de generalización.

Para la evaluación definitiva del modelo se utilizó el conjunto de datos de prueba, obteniendo una precisión global del 97%. El desglose detallado del rendimiento por cada categoría de calidad del aire se presenta en la Tabla 17

Tabla 17

Desempeño global de predicción

Clase	Precisión	Recall	F1-Score	SopORTE
Limpio (0)	0.97	0.98	0.98	237
Elevado (1)	0.98	0.95	0.96	217
Insalubre (2)	0.98	1.00	0.99	200
Promedio Macro	0.98	0.98	0.98	654
Promedio Ponderado	0.98	0.98	0.98	654

El análisis métrico revela un desempeño altamente equilibrado, avalado por promedios globales de 0.98. Destaca la clase Insalubre con un recall de 1.00, garantizando la detección total de eventos críticos. La categoría Limpio mostró gran solidez, mientras que la

clase Elevado presentó una leve reducción en el recall (0.95), atribuible a solapamientos en los límites de decisión con otras categorías. El hecho de que estas métricas se mantengan altas frente a un soporte considerable de 654 muestras valida la robustez de la solución propuesta para su implementación en escenarios reales.

3.7.1.7. Conversión de modelo a TFLite y exportación de escalador.

Con el objetivo de implementar el modelo en dispositivos con recursos limitados como es el Raspberry Pi Zero 2W, se procedió a la conversión del modelo entrenado a formato TensorFlow Lite (TFLite). Esta conversión permite optimizar el tamaño y la eficiencia del modelo, facilitando su ejecución en entornos embebidos, como microcontroladores o dispositivos móviles, sin comprometer significativamente su rendimiento de predicción.

El proceso de conversión se realizó a partir del modelo original en formato TensorFlow (.h5), utilizando la API de conversión de TFLite provista por TensorFlow como se muestra en la figura 33. Durante esta etapa se mantuvo la precisión del modelo en punto flotante (float32), aunque el entorno permite exploraciones futuras con técnicas de cuantización para una mayor optimización.

Figura 33

Conversión del modelo a tflite.

```

import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd

# Ruta del archivo del modelo
path = "modelo"
# Cargar el modelo de Keras
modelo = tf.keras.models.load_model(path + ".h5")

converter = tf.lite.TFLiteConverter.from_keras_model(modelo)
tflite_modelo = converter.convert()

# Guardar el modelo convertido en un archivo .tflite
with open(path + '.tflite', 'wb') as f:
    f.write(tflite_modelo)
print("\r\nTensorFlow Keras H5 model converted to a TensorFlow Lite model!\r\n")

```

Adicionalmente, se exportó el objeto escalador (*Scaler*) utilizado en la etapa de preprocesamiento de datos. Esta exportación es crucial para garantizar que los datos de entrada durante la inferencia mantengan la misma escala que los datos utilizados durante el entrenamiento, asegurando así la coherencia del desempeño del modelo. El escalador fue almacenado en formato binario mediante la biblioteca *joblib*, lo que permite su carga eficiente en la fase de despliegue como se muestra en la figura 34.

Figura 34

Exportación del escalador utilizado.

```

import joblib
joblib.dump(scaler, 'scaler.pkl')

```

3.7.2. Implementación del software.

La implementación del software se llevó a cabo conforme a la arquitectura funcional definida en la sección 3.6.2. El proceso se inicia con la adquisición de datos ambientales, la cual es ejecutada por el microcontrolador Arduino Mega Pro Mini 2560, encargado de recopilar las señales provenientes de los sensores conectados al sistema. Una vez capturada la

información, esta es transmitida para su procesamiento en tiempo real al Raspberry Pi Zero 2W, permitiendo así la interpretación inmediata de las variables medidas.

3.7.2.1. Programación Lectura y procesamiento de datos en Arduino Mega

El código de lectura desarrollado en el entorno Arduino IDE empleó bibliotecas específicas para la gestión de sensores mediante el protocolo I2C, el sensor de ozono (0x73), el sensor de NO₂ (0x74) y el BME280 (0x77) optimizando el uso de pines. En la sección de configuración (setup()), se inicializan los sensores de ozono (DFRobot_OzoneSensor), dióxido de nitrógeno (DFRobot_MultiGasSensor) y el módulo BME280 para parámetros ambientales para asegura la robustez del sistema ante fallos de conexión. Y se realiza coloca un tiempo de calentamiento de 3 minutos como se muestra en la figura 35.

Figura 35

Inicialización de sensores y calentamiento.

```
while (!gas.begin()) {
  Serial.println("IIC NO2 Sensor is not found!");
  delay(1000);
}
Serial.println("IIC NO2 Sensor is connected successfully!");
gas.changeAcquireMode(gas.PASSIVITY);
delay(1000);
gas.setTempCompensation(gas.ON);
// Inicializar BME280
bme.reset();
while (bme.begin() != BME::eStatusOK) {
  Serial.println("BME280 initialization failed");
  printLastOperateStatus(bme.lastOperateStatus);
  delay(2000);
}
Serial.println("BME280 initialized successfully");
// Calentamiento
timer = millis();
Serial.print("Calentamiento: ");
while (millis() - timer < 180000) {
  if (millis() - timer > 1000) {
    Serial.print("*");
  }
}
heating = false;
Serial.println("\nCalentamiento completed.");
```

La adquisición de datos se realiza mediante la función `collect_air_quality_data()`, se encarga de realizar la lectura de sensores en tiempo real, estructurar los datos en formato JSON y enviarlos a través del puerto serial hacia la Raspberry Pi Zero 2 W para su posterior procesamiento como se muestra en la figura 36.

Durante cada iteración, se recolectan datos provenientes de múltiples sensores:

- **NO₂**: Utilizando un sensor de gases que devuelve la concentración en partes por millón (ppm), accedida mediante `readGasConcentrationPPM()`.
- **Ozono (O₃)**: Obtenido con `readOzoneData()` del sensor correspondiente, utilizando un parámetro de agregación definido por `COLLECT_NUMBER`.
- **Temperatura y humedad**: Leídas desde un sensor BME280, utilizando sus respectivos métodos de acceso: `getTemperature()`, `getPressure()`, `calAltitude()`, y `getHumidity()`.
- **GPS**: Recolecta las variables de latitud, Norte o Sur, longitud y este o oeste necesarios para realizar el posicionamiento exacto del dispositivo.

Entre cada lectura se inserta un retardo (`delay(500)` milisegundos) para evitar lecturas inestables y garantizar la correcta adquisición de los datos.

Tras la adquisición de las variables, se construye manualmente una cadena en formato JSON que contiene los valores de NO₂, O₃, temperatura y humedad, junto con las coordenadas geográficas del dispositivo (lat y lon). Para ello, se emplea concatenación de cadenas a partir de los valores numéricos formateados con una precisión decimal controlada (por ejemplo, `String(valor, 2)` para dos decimales). Finalmente, la cadena JSON construida es enviada a través del puerto serial (`Serial.print(json)` o equivalente), lo que permite que el sistema receptor (en este caso, la Raspberry Pi) escuche y procese los datos de forma continua. Esta transmisión se realiza en formato de texto plano, lo que facilita su interpretación mediante bibliotecas estándar de decodificación JSON en Python.

Figura 36

Función de recolección de datos y archivo JSON.

```
void collect_air_quality_data() {
    // Generar datos simulados
    no2Concentration = gas.readGasConcentrationPPM();
    delay(500);
    ozoneConcentration = Ozone.readOzoneData(COLLECT_NUMBER);
    delay(500);
    temperature = bme.getTemperature();
    pressure = bme.getPressure();
    altitude = bme.calAltitude(SEA_LEVEL_PRESSURE, pressure);
    humidity = bme.getHumidity();
    delay(500);

    // Construir string JSON manualmente
    String json = "{";
    json += "\"NO2\": " + String(no2Concentration, 2) + ",";
    json += "\"O3\": " + String(ozoneConcentration) + ",";
    json += "\"temp\": " + String(temperature, 2) + ",";
    json += "\"hum\": " + String(humidity, 2) + ",";
    json += "\"lat\": " + String(latitude, 4) + ",";
    json += "\"lon\": " + String(longitude, 4);
    json += "}";

    // Enviar por Serial
    Serial.println(json);
}
```

3.7.2.2. Procesamiento de Raspberry Pi Zero 2W

Para el sistema operativo del dispositivo, se utilizó la imagen 2023-02-21-raspbros-bullseye-armhf-lite.img.xz, una versión ligera de Raspberry Pi OS basada en Debian Bullseye. Esta elección se debió a su bajo consumo de recursos y compatibilidad con la Raspberry Pi Zero 2W, lo cual permitió un entorno eficiente y estable para la ejecución de los procesos de adquisición de datos y comunicación del sistema.

- **Importación de bibliotecas y configuración de entorno**

El sistema hace uso de bibliotecas especializadas como serial para la comunicación con el Arduino vía UART, json para el manejo de estructuras de datos, pandas y numpy para el procesamiento de datos, y joblib para cargar el modelo de escalamiento previamente entrenado. Adicionalmente, se utiliza tflite_runtime.interpreter para ejecutar el modelo ligero optimizado para entornos embebidos.

También se configura la conexión con Firebase Firestore a través de la biblioteca `firebase_admin`, utilizando un certificado de autenticación que permite el acceso seguro a la base de datos remota.

Figura 37

Importación de librerías.

```
import serial
import json
import pandas as pd
import joblib
import numpy as np
import time
from tflite_runtime.interpreter import Interpreter
import firebase_admin
from firebase_admin import credentials, firestore
from sklearn.preprocessing import MinMaxScaler
```

- Inicialización de Firebase y carga del modelo

Se inicializa la conexión con Firebase mediante una credencial en formato `.json` y se carga el objeto `scaler` (entrenado previamente con `MinMaxScaler`) junto con el modelo en formato `.tflite`. Este último es interpretado utilizando `Interpreter` de TensorFlow Lite.

Figura 38

Credenciales y carga de datos.

```
# Inicializar Firebase
cred = credentials.Certificate("tesis-aiq-firebase-adminsdk-fbsvc-5a31bf1149.json")
firebase_admin.initialize_app(cred)
db = firestore.client()
# Cargar scaler y modelo
scaler = joblib.load("scaler_entrenado2.pkl")
interpreter = Interpreter(model_path="modelo_96_test0_52.tflite")
interpreter.allocate_tensors()
# Entradas y salidas
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()
```

- Lectura de datos y ejecución de inferencias

A través del puerto serial (`/dev/serial0`) se escuchan los datos enviados desde el Arduino. Cada línea es interpretada como una estructura JSON que contiene las variables ambientales: dióxido de nitrógeno (NO₂), ozono (O₃), temperatura y humedad.

Si el paquete recibido contiene todos los campos esperados, los datos son convertidos en un DataFrame de pandas, escalados mediante el objeto scaler y posteriormente convertidos en una matriz de entrada para el modelo neuronal.

Figura 39

Procesamiento de datos.

```
try:
    while True:
        line = ser.readline().decode('utf-8').strip()
        if line:
            try:
                data = json.loads(line)
                print("JSON recibido:", data)

                if all(k in data for k in ['NO2', 'O3', 'temp', 'hum']):
                    df = pd.DataFrame([
                        {'no2': data['NO2'],
                         'ozone': data['O3'],
                         'temperature': data['temp'],
                         'humidity': data['hum']}
                    ])

                    # Escalar
                    scaled = scaler.transform(df)
                    scaled_dict = dict(zip(df.columns, scaled[0]))
                    print("Escalado:", scaled_dict)

                    # Preparar entrada al modelo
                    input_data = np.array([list(scaled_dict.values())], dtype=np.float32)
                    interpreter.set_tensor(input_details[0]['index'], input_data)
                    interpreter.invoke()
                    output_data = interpreter.get_tensor(output_details[0]['index'])
                    predicted_class = int(np.argmax(output_data))
                    print(f"📡 Predicción (clase): {predicted_class}")
```

- Registro de resultados y envío a Firestore

Una vez obtenida la predicción, se serializan los datos a formato JSON con los valores originales, la clase predicha y una marca temporal. Si el paquete de datos incluye coordenadas geográficas (lat, lon), se adjunta un GeoPoint compatible con Firebase. Estos datos se envían a un servidor intermedio que recibirá y realizará la publicación en la base de datos de Firestore.

Figura 40

Envío de datos a Servidor.

```

# 1 Inicialización base
send_at("AT")
send_at("AT+CFUN=1")
send_at("AT+CPIN?")
send_at("AT+CSQ")
send_at("AT+CREG?")

# 2 Configuración del GPRS
send_at('AT+SAPBR=3,1,"CONTTYPE","GPRS"')
send_at(f'AT+SAPBR=3,1,"APN","{APN}"')
send_at("AT+SAPBR=1,1", wait=5)
send_at("AT+SAPBR=2,1")

# 3 Configuración HTTP
send_at("AT+HTTPTERM", show=False)
send_at("AT+HTTINIT")
send_at('AT+HTTTPARA="CID",1')
send_at(f'AT+HTTTPARA="URL","http://{SERVER}{RESOURCE}"')
send_at('AT+HTTTPARA="CONTENT","application/json"')

# 4 Cargar y enviar JSON
json_length = len(JSON_DATA.encode("utf-8"))
send_at(f"AT+HTTPDATA={json_length},10000")
time.sleep(1)
ser.write(JSON_DATA.encode("utf-8"))
time.sleep(3)
print(f">>> Enviado JSON limpio ({json_length} bytes)")

# 5 Ejecutar POST HTTP
reply = send_at("AT+HTTTPACTION=1", wait=10)

# Si el servidor responde 200 (OK)
if "+HTTTPACTION: 1,200" in reply:
    response = send_at("AT+HTTTPREAD", wait=5)
    print("\nⓂ Respuesta del servidor:\n", response)
else:
    print("\nⓂ Error o sin respuesta HTTP válida:\n", reply)

```

El servidor está desarrollado en Node.js utilizando el framework Express y el SDK de Firebase Admin para la conexión con Firestore levantado de manera local. Su función principal es recibir datos enviados desde el modulo SIM800L o mediante solicitudes CURL, procesarlos y almacenarlos directamente en la base de datos Firestore como se muestra en la figura 41.

Figura 41

Recolección de datos del servidor.

```

// Crear documento
const data = {
  humedad: parseFloat(humedad),
  no2: parseFloat(no2),
  o3: parseFloat(o3),
  prediccion: parseInt(prediccion),
  temperatura: parseFloat(temperatura),
  timestamp: parseInt(timestamp),
  ubicacion: ubicacion,
};
// Guardar en Firestore con timestamp como ID
await db.collection("registros").doc(String(timestamp)).set(data);
console.log("✅ Datos guardados correctamente:", data);
res.status(200).json({
  // mensaje: "✅ Datos guardados en Firestore con GeoPoint"
  mensaje: "1"
  // datos_recibidos: data,
});
} catch (err) {
  console.error("❌ Error al guardar:", err);
  res.status(500).send("Error al guardar en Firestore: " + err);
}
});

```

3.7.2.3. Almacenamiento de Datos en Firestore.

Como parte del sistema de monitoreo, se integró una base de datos en la nube utilizando *Firestore*, un servicio de base de datos NoSQL provisto por Google Cloud. Esta implementación permite almacenar, consultar y sincronizar los registros ambientales registrados desde Raspberry Pi Zero 2W de forma segura y escalable.

Cada registro almacenado en *Firestore* representa una observación puntual capturada por el sistema y está estructurado como un documento dentro de la colección denominada registros. Los documentos se identifican mediante un ID único basado en la marca de tiempo UNIX (*timestamp*), lo cual garantiza un orden cronológico y evita conflictos por duplicación.

Estructura del documento

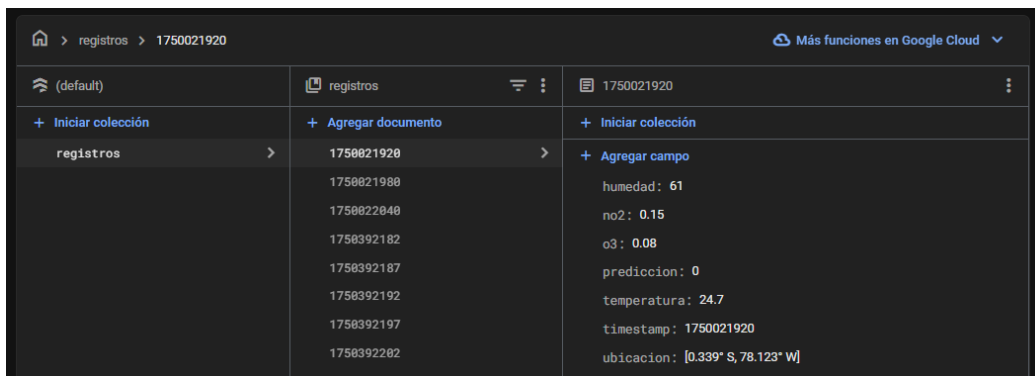
La figura 42 muestra un ejemplo de documento con ID 1750021920, el cual contiene los siguientes campos:

- humedad: *Number*, representa el porcentaje de humedad relativa registrada, en este caso 61%

- Dióxido de Nitrógeno (NO₂): *Number*, concentración de dióxido de nitrógeno medida en partes por millón (ppm), valor 0.15
- Ozono (O₃): *Number*, concentración de ozono en ppm, con un valor de 0.08
- temperatura: *Number*, medición de la temperatura ambiente en grados Celsius, reportando 24.7°C.
- predicción: *Number*, resultado del modelo de inferencia basado en red neuronal. En este caso, el valor 0 representa una clase específica (por ejemplo, aire limpio)
- *timestamp*: *Number*, marca de tiempo que indica el momento exacto de la recolección, usado también como ID del documento.
- ubicación: *Geopoint*, coordenadas geográficas del dispositivo al momento de la medición, expresadas en formato decimal con dirección cardinal

Figura 42

Estructura de datos en firestore.



3.7.2.4. Construcción de Interfaz Web

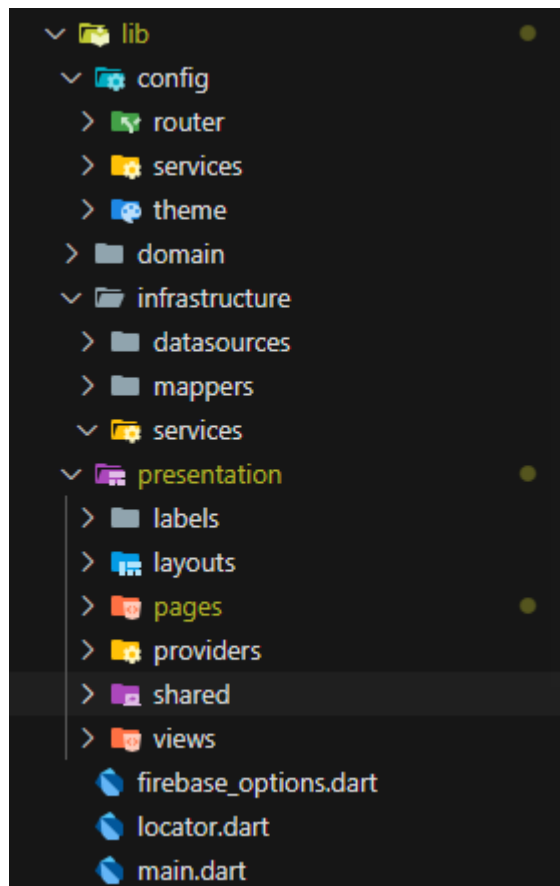
Se desarrolló una interfaz web utilizando Flutter, con el objetivo de visualizar en tiempo real los datos recolectados, las predicciones generadas por el modelo de inteligencia artificial y la ubicación geográfica de las mediciones. El desarrollo se basó en los principios de Clean Architecture, favoreciendo la mantenibilidad, escalabilidad y separación de responsabilidades dentro del proyecto.

- **Estructura del Proyecto**

La organización del código se divide en capas bien definidas dentro del directorio lib, siguiendo una arquitectura de tipo “vertical”, que promueve la independencia entre módulos como se muestra en la figura 43.

Figura 43

Organización del proyecto.



El proyecto se organiza en capas para separar responsabilidades. El directorio config centraliza rutas, servicios globales y estilos visuales. El módulo domain encapsula la lógica de negocio pura, mientras que infrastructure gestiona la persistencia de datos y la comunicación con Firestore. Finalmente, la capa presentation organiza la interfaz de usuario distribuyendo los elementos en etiquetas de texto, estructuras de diseño comunes y las páginas de navegación. El estado de la aplicación se maneja mediante los elementos presentes en *providers*, mientras que *shared/widgets* concentra componentes visuales reutilizables

como tarjetas informativas, botones y gráficos. Adicionalmente, *views* alberga fragmentos específicos que forman parte de las páginas, como por ejemplo la vista de los sensores activos o la visualización georreferenciada de datos.

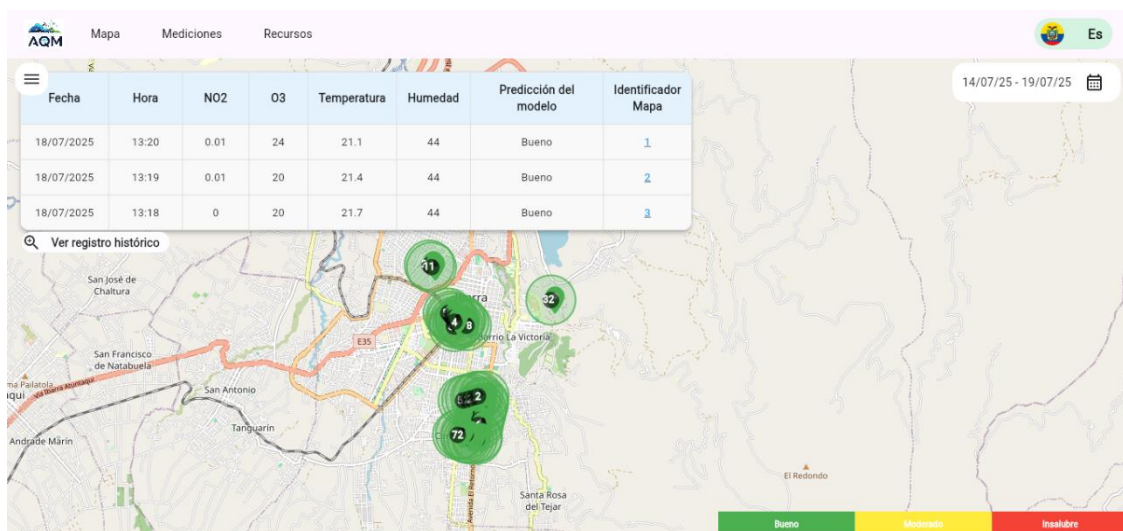
- **Pantalla principal**

En la figura 44 se presenta la pantalla principal de la página web desarrollada donde se presenta un menú desplegable accesible mediante un ícono de hamburguesa, acompañado del logotipo del sistema "AQM" y un encabezado de navegación con las secciones: "Mapa", "Mediciones" y "Recursos". Estas opciones permiten acceder rápidamente a las distintas funcionalidades del sistema. La interfaz cuenta también con un botón de configuración ubicado en la esquina superior derecha.

Justo debajo del menú, se despliega una tabla que muestra los registros ambientales obtenidos. Esta tabla contiene columnas para la Fecha, Hora, NO₂, O₃, Temperatura, Humedad, la Predicción del modelo y un Identificador del mapa.

Figura 44

Pantalla principal de interfaz web.



- **Pantalla de resultados**

En la figura 45, se muestra la pantalla de mediciones del dispositivo donde se muestra un registro histórico de los datos por día y filtrados por fecha actual. Su propósito principal es

presentar de forma organizada y detallada los registros de calidad del aire capturados por los sensores en diferentes ubicaciones.

Figura 45

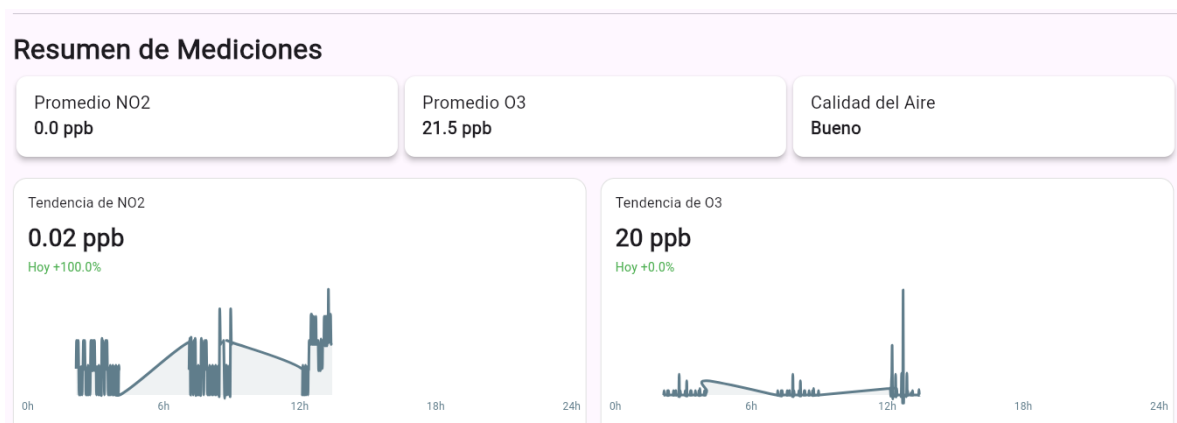
Pantalla de mediciones de interfaz web.

Ubicación	Fecha	Hora	Temperatura	NO2	O3	Calidad del Aire
Ibarra_Imbabura (0.346135691850259,-78.1209552436243)	14/07/2025	02:05	24.1 °C	0.01 ppb	26 ppb	Bueno
Ibarra_Imbabura (0.346135691850259,-78.1210115700116)	14/07/2025	02:06	24.3 °C	0 ppb	26 ppb	Bueno
Ibarra_Imbabura (0.346134584986391,-78.1209127220369)	14/07/2025	02:07	24.3 °C	0.01 ppb	26 ppb	Bueno
Ibarra_Imbabura (0.346071034148154,-78.1209630340343)	14/07/2025	02:08	22.0 °C	0.01 ppb	20 ppb	Bueno
Ibarra_Imbabura (0.346221967388298,-78.120948470035)	14/07/2025	02:09	21.8 °C	0 ppb	20 ppb	Bueno
Ibarra_Imbabura (0.346135691850259,-78.1209552436243)	14/07/2025	02:10	21.1 °C	0.01 ppb	32 ppb	Bueno
Ibarra_Imbabura (0.346135691850259,-78.1210115700116)	14/07/2025	02:11	21.1 °C	0 ppb	40 ppb	Bueno

En la Figura 46 se presenta un resumen filtrado por día que permite visualizar de manera consolidada el estado general de la calidad del aire y los valores promedio de las variables medidas por los sensores. Esta vista ofrece una representación simplificada y accesible de los datos recolectados durante una jornada específica, lo que facilita su interpretación.

Figura 46

Pantalla de mediciones resumen.



Toda la información relacionada con el código desarrollado se encuentra disponible en el repositorio de GitHub:

https://github.com/FirstRSSO/Sistema_IoT_Monitoreo_NO2_O3_UTN

3.7.3 Implementación del hardware.

En cuanto a la implementación del hardware, se llevó a cabo la integración de los componentes mediante una conexión directa al Arduino Mega Mini 250 Pro, tal como se detalló en la sección 3.6.4. Debido a esta configuración, no fue necesario diseñar una placa de circuito impreso (PCB), por lo que se optó por una adaptación directa de cada componente al interior del case.

3.7.3.1 Fuente del dispositivo.

Con base en el análisis de consumo energético presentado en la sección 3.6.8, se determinó la necesidad de una fuente de alimentación capaz de suministrar una corriente continua de al menos 737 mA durante un período operativo estimado de 8 horas. En función de estos requerimientos, se seleccionó una batería externa (*power bank*) de alta capacidad, modelo WGP de 10 400 mAh y salida de 25 W, la cual proporciona una tensión de salida de 5 V compatible con todos los componentes del sistema como se muestra en la figura 47.

Esta elección garantiza no solo la autonomía energética necesaria, sino también una reserva adicional de capacidad, lo que permite compensar posibles pérdidas por conversión, eficiencia del regulador interno de la batería y variaciones en la demanda energética durante el funcionamiento real.

Figura 47

Batería externa WGP de 10 400 mAh.



3.7.3.2. Dispositivo Armado

El diseño interno del dispositivo se ha organizado en dos niveles funcionales o capas, con el objetivo de optimizar el espacio disponible y asegurar una distribución eficiente de los componentes electrónicos.

En la primera capa, se ubican los elementos principales de procesamiento y alimentación, específicamente la batería externa, la Raspberry Pi Zero 2W y el Arduino Mega Pro Mini 2560. Debido a que la Raspberry Pi Zero 2W opera exclusivamente con señales lógicas de 3.3 V en sus pines GPIO, mientras que el microcontrolador Arduino utiliza niveles lógicos de 5 V, fue necesario integrar un convertidor de voltaje bidireccional para permitir una comunicación segura entre ambos dispositivos, tal como se ilustra en la Figura 48. Las conexiones internas se realizaron mediante cables tipo Molex, lo cual permite una integración ordenada y facilita la conexión posterior de los sensores al microcontrolador.

Figura 48

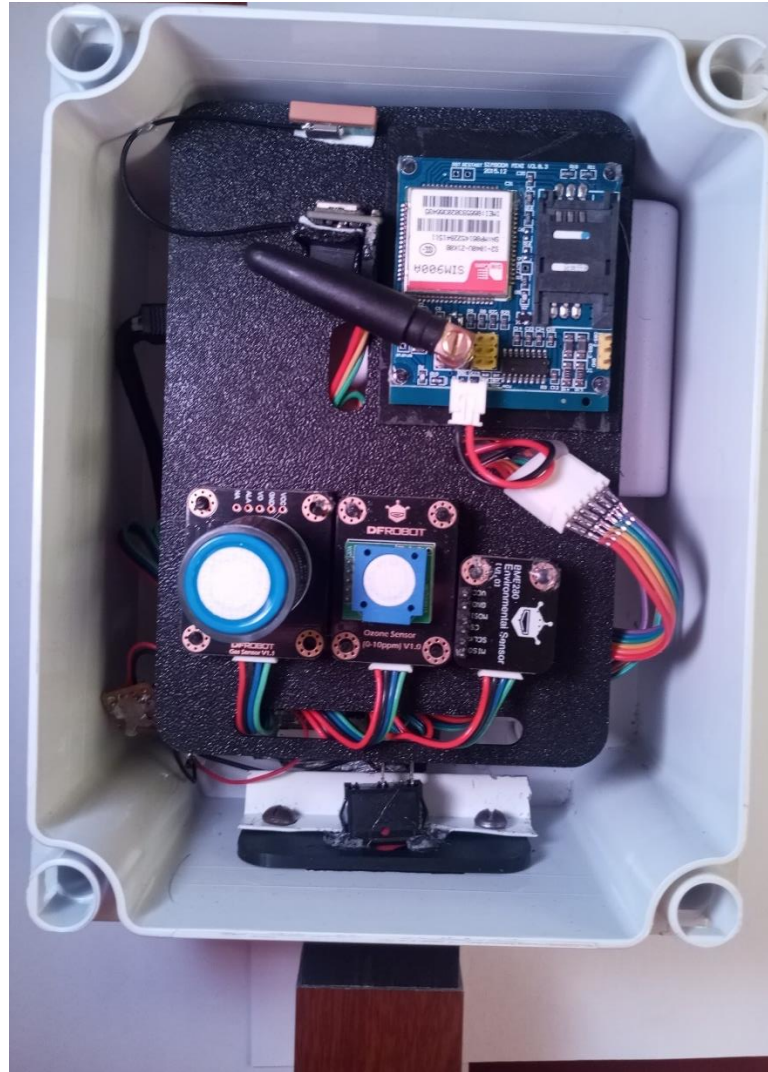
Primera capa interna del dispositivo.



La segunda capa, mostrada en la Figura 49, alberga exclusivamente los sensores ambientales, los cuales se encuentran estratégicamente distribuidos para facilitar su conexión directa al microcontrolador. Esta disposición permite un acceso rápido para tareas de mantenimiento o reemplazo, así como una mejor gestión del cableado interno.

Figura 49

Segunda capa interna del dispositivo.



En cuanto al diseño externo, y como se aprecia en la Figura 50, la carcasa frontal fue sellada en sus bordes utilizando silicona fría, con el fin de garantizar la impermeabilidad del dispositivo y proteger los componentes internos frente a condiciones ambientales adversas. Adicionalmente, se incorporó un refuerzo metálico en la parte frontal, destinado a facilitar la fijación segura del dispositivo sobre un trípode estándar, asegurando así su estabilidad durante las sesiones de monitoreo en campo.

Figura 50

Parte frontal del dispositivo.



CAPÍTULO IV: PRUEBAS Y RESULTADOS

Este capítulo corresponde a la cuarta fase de la metodología Bottom-Up, orientada a la verificación integral del sistema desarrollado. En esta etapa se analizan los diferentes componentes que conforman la solución propuesta, iniciando con la evaluación individual de los módulos internos, la eficacia del sistema de protección y el desempeño de la plataforma web destinada a la visualización de datos.

Asimismo, se definen y asignan puntos estratégicos de prueba en función de criterios representativos del entorno urbano, en los cuales se valida el sistema de monitoreo ambiental mediante la recolección de datos en tiempo real. Este procedimiento permite comprobar el correcto funcionamiento del dispositivo bajo condiciones reales de operación.

4.1 Pruebas del sistema.

Para cada prueba se establecieron criterios de validación específicos que permiten determinar de manera objetiva si el sistema cumple con los parámetros de desempeño esperados. El proceso inició con la ejecución de pruebas unitarias sobre cada componente de manera aislada, orientadas a verificar su correcto funcionamiento interno. Posteriormente, se realizaron pruebas de integración que evaluaron el desempeño conjunto de los módulos frente a diferentes escenarios de operación.

La Tabla 18 muestra las abreviaturas de las pruebas realizadas usando un prefijo con su componente para su identificación.

Tabla 18

Nomenclatura de pruebas

ABREVIATURA	SIGNIFICADO
CAL	Calibración de sensores
PR-S	Prueba de sensores
PR-COM	Prueba de comunicación serial

PR-PROC	Prueba de procesamiento en la Raspberry Pi
PR-INT	Prueba de internet
PR-CASE	Prueba de case
PR-DC	Prueba de sistema completo

En la Tabla 19 se muestra las descripciones de cada una de las pruebas junto con los resultados esperados para su evaluación.

Tabla 19

Pruebas realizadas para la validación del sistema.

PRUEBAS INDIVIDUALES		
ID	DESCRIPCIÓN	RESULTADO ESPERADO
CAL	Ejecución de calibración de los sensores de NO ₂ y O ₃ .	Recolección de datos estable
PR-S01	Se verifica el encendido de los sensores y estabilidad de resultados transcurrido el calentamiento.	Los sensores mostrar el led de encendido y estabilidad en los datos.
PR-S02	Se verifica la detección de variables acordes a un ambiente libre de exposiciones directas.	Concentraciones acordes al entorno.
PR-S03	Se verifica la detección de variables acorde a una exposición directa de fuente	Concentraciones ligeramente elevadas de acuerdo a la exposición.
PR-COM	Se comprueba que el Arduino Mega recopile los datos de los sensores y los	La Raspberry Pi recibe los datos en el formato y tiempo esperado, sin errores de transmisión.

	envíe correctamente a la Raspberry Pi sin pérdida de información.	
PR- PROC	Se valida la capacidad de la Raspberry Pi 2W para procesar los datos recibidos y ejecutar la red neuronal, generando salidas interpretables.	Los datos son procesados correctamente y la red neuronal produce inferencias en tiempo real.
PR- INT	Se verifica la conexión estable de la placa SIM800L v2 y la transmisión de datos hacia la nube.	La conexión a internet se mantiene estable y los datos son enviados correctamente a la base de datos.
PR- CASE	Se evalúa la resistencia física y la ventilación del case con aberturas, asegurando protección de los componentes sin comprometer el flujo de aire.	Los componentes permanecen protegidos frente a climas diferentes.
PR- WEB	Se revisa el funcionamiento de la plataforma en Netlify y su integración con la base de datos Firestore, asegurando que los datos se almacenen y se visualicen en tiempo real.	Los datos son almacenados correctamente en Firestore y mostrados en la web sin retrasos significativos

PRUEBAS COMPLETAS DEL SISTEMA

ID	DESCRIPCIÓN	RESULTADO ESPERADO
PR- DC00	Se valida la integración total del sistema en condiciones reales, verificando la captura, procesamiento, transmisión y visualización de datos ambientales.	Todos los componentes funcionan de manera conjunta y los datos se presentan correctamente en la plataforma web.

PR-DC01	Se envía intencionalmente un valor fuera de rango para verificación del manejo de error	El sistema descarta o reporta el dato invalido sin afectar le procesamiento
PR-DC02	Se interrumpe la red móvil durante 3 min.	El sistema reintenta conexión y reanuda transmisión una vez restablecida la señal.
PR-DC03	Se envían múltiples tramas de datos en ráfagas rápidas desde Arduino.	El sistema mantiene la integridad de los datos sin pérdida ni colapso del procesamiento
PR-DC04	Funcionamiento continuo durante 24 y verificación de duración de energía.	Autonomía estable, sin sobrecalentamiento ni fallas de hardware/software
PR-DC05	Visualización en la plataforma desde un <i>smarthphone</i> .	Acceso fluido y datos en tiempo real sin retraso significativo.

4.1.2 Pruebas Individuales de componentes del sistema.

Con base en la Tabla 19, se llevaron a cabo las pruebas de funcionamiento de los diferentes componentes que integran el sistema.

4.1.2.1 Calibración de los sensores (CAL)

Antes de realizar las pruebas de validación del sistema, se consideró la calibración de los sensores empleados:

- Sensor Gravity Electrochemical Nitrogen Dioxide (NO₂, 0–20 ppm, I2C/UART)
- Sensor Gravity Electrochemical Ozone (O₃, 0–10 ppm, I2C)

DFrobot(2023, n.d) describe en la documentación oficial que los sensores electroquímicos de la serie *Gravity* cuentan con calibración de fábrica, realizada bajo condiciones controladas por el fabricante, lo que garantiza que las salidas eléctricas correspondan a concentraciones conocidas de cada gas dentro de su rango de operación. Sin embargo, siguiendo las recomendaciones del fabricante se realiza un precalentado del sensor durante 24 horas por primera vez y debido a que se trata de sensores electroquímicos se debe calentar durante 5 minutos para el uso durante cada encendido.

En el caso del BME280, no se requiere calibración adicional, ya que el fabricante proporciona compensaciones internas para temperatura, humedad y presión, verificadas con mediciones de referencia en campo abierto.

De esta manera, se establece que los sensores después del precalentado operan dentro de los rangos esperados y están aptos para las pruebas posteriores.

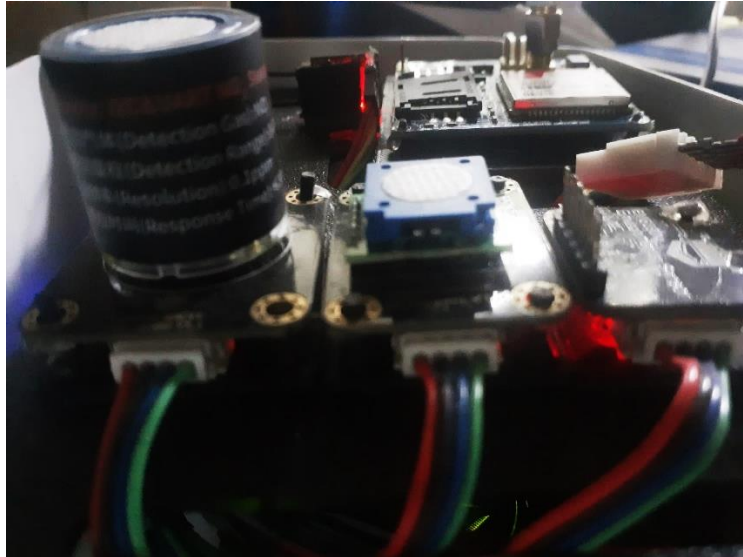
4.1.2.2 Encendido de sensores (PR-S01)

Con el propósito de validar el correcto arranque de los módulos de medición, se verificó el encendido de los sensores de NO₂, O₃ y BME280, así como la estabilidad de sus lecturas tras el tiempo de calentamiento recomendado por el fabricante de cinco minutos.

En primera instancia, se constató la activación visual de los sensores electroquímicos mediante el LED indicador de estado, confirmando el suministro de energía y la correcta inicialización de cada módulo. Del mismo modo como se establece en las hojas técnicas, se realiza el precalentamiento de los sensores electroquímicos y se verifico estabilidad de los datos como se muestra en la figura 51.

Figura 51

Sensores encendidos



4.1.2.3 Exposición de sensores ambiente libre de exposiciones (PR-S02)

Posteriormente de la comprobación de encendido correcto y calibración inicial, se evaluó la estabilidad proporcionalmente al entorno. A modo experimental, se tomaron mediciones en un campo abierto, donde las concentraciones de gases y las condiciones ambientales permanecieron dentro de rangos bajos y estables, acordes al entorno, como se muestra en la Figura 52.

Figura 52

Prueba de sensores en campo abierto dentro del case.



4.1.2.4 Exposición directa de sensores a fuente de gases (PR-S03)

En contraste con la prueba anterior, se realizó una prueba de exposición directa al tubo de escape de un automóvil en funcionamiento. Cabe mencionar que los motores de combustión emiten principalmente monóxido de nitrógeno (NO), el cual en presencia de oxígeno atmosférico se oxida rápidamente a dióxido de nitrógeno (NO₂). Este procedimiento permitió registrar variaciones significativas en los valores medidos por los sensores, evidenciando su capacidad de respuesta frente a escenarios de alta concentración, como se presenta en la figura 53.

Figura 53

Prueba de sensores en fuente directa de NO_x dentro del case.



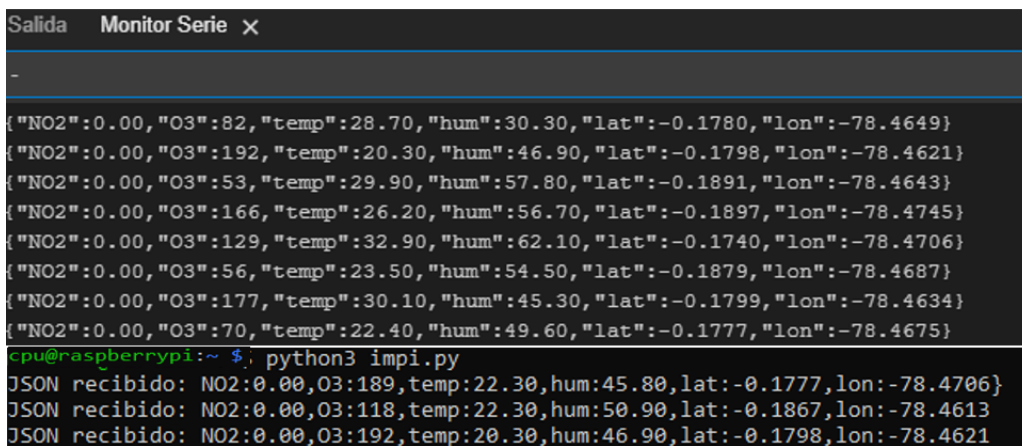
4.1.2.5 Prueba de comunicación serial (PR-COM)

Una vez comprobada la recolección de datos por parte de los sensores, se procedió a la validación de la comunicación serial UART, utilizada entre Arduino Mega y la Raspberry Pi Zero 2W. El propósito de esta prueba fue garantizar la transmisión íntegra de la información capturada por los sensores hacia la unidad de procesamiento, evitando pérdidas o distorsiones en los datos. Para ello, se enviaron diferentes tramas de prueba en formato

estructurado JSON, verificando que los valores recibidos en la Raspberry Pi coincidieran de manera exacta con los generados en el Arduino. Los resultados confirmaron la estabilidad y confiabilidad de la comunicación, asegurando la correcta disponibilidad de los datos para su posterior procesamiento, como se muestra en la figura 54.

Figura 54

Recepción de datos de Arduino a Raspberry Pi Zero 2W.



```
Salida  Monitor Serie x
-
{"NO2":0.00,"O3":82,"temp":28.70,"hum":30.30,"lat":-0.1780,"lon":-78.4649}
{"NO2":0.00,"O3":192,"temp":20.30,"hum":46.90,"lat":-0.1798,"lon":-78.4621}
{"NO2":0.00,"O3":53,"temp":29.90,"hum":57.80,"lat":-0.1891,"lon":-78.4643}
{"NO2":0.00,"O3":166,"temp":26.20,"hum":56.70,"lat":-0.1897,"lon":-78.4745}
{"NO2":0.00,"O3":129,"temp":32.90,"hum":62.10,"lat":-0.1740,"lon":-78.4706}
{"NO2":0.00,"O3":56,"temp":23.50,"hum":54.50,"lat":-0.1879,"lon":-78.4687}
{"NO2":0.00,"O3":177,"temp":30.10,"hum":45.30,"lat":-0.1799,"lon":-78.4634}
{"NO2":0.00,"O3":70,"temp":22.40,"hum":49.60,"lat":-0.1777,"lon":-78.4675}
cpu@raspberrypi:~ $ python3 impi.py
JSON recibido: NO2:0.00,O3:189,temp:22.30,hum:45.80,lat:-0.1777,lon:-78.4706}
JSON recibido: NO2:0.00,O3:118,temp:22.30,hum:50.90,lat:-0.1867,lon:-78.4613
JSON recibido: NO2:0.00,O3:192,temp:20.30,hum:46.90,lat:-0.1798,lon:-78.4621
```

4.1.2.6 Prueba de procesamiento (PR-PROC)

Posteriormente a la verificación de la estabilidad en la comunicación y correcta recepción de los datos, se procedió a la validación del procesamiento en la Raspberry Pi Zero 2W. Esta etapa consistió en comprobar la capacidad del dispositivo para ejecutar los algoritmos de preprocesamiento y la red neuronal implementada, asegurando que las entradas provenientes de los sensores fueran interpretadas de manera adecuada. Para ello, se analizaron los tiempos de respuesta, y la coherencia de los resultados obtenidos frente a los datos recibidos. Los ensayos demostraron que la Raspberry Pi cumplió con el procesamiento requerido en tiempo real, sin presentar fallos de ejecución ni retrasos significativos como se muestra en la figura 55.

Figura 55

Datos procesados por Raspberry Pi Zero 2W.

```

cpu@raspberrypi:~ $ tail -f registros.csv
1753531248,0.29,82.0,30.4,25.2,0,-0.1807,-78.4678
1753531311,0.2,20.0,30.1,24.6,0,-0.1807,-78.4678
1753531374,0.29,20.0,29.8,26.3,0,-0.1807,-78.4678
1753531437,0.2,20.0,29.9,25.2,0,-0.1807,-78.4678
1753531500,0.29,20.0,29.9,25.0,0,-0.1807,-78.4678
1753531563,0.2,61.0,30.1,25.5,0,-0.1807,-78.4678
1753531626,0.29,20.0,30.3,28.8,0,-0.1807,-78.4678
1753531689,0.29,126.0,30.6,28.6,1,-0.1807,-78.4678
1753531753,0.29,23.0,30.8,27.8,0,-0.1807,-78.4678
1753531816,0.29,20.0,31.0,27.9,0,-0.1807,-78.4678

```

4.1.2.7 Prueba de internet (PR-INT)

Para el envío de datos hacia la base de datos en la nube, se realizó en primera instancia la verificación de la conectividad del sistema mediante la placa SIM800L v2. Esta prueba consistió en comprobar el establecimiento de la red móvil y la transmisión de información hacia un servidor intermedio que, a través de peticiones HTTP, recibe los datos y los reenvía posteriormente a la base de datos *Firestore*, tal como se mencionó en la sección anterior. Los resultados de esta validación se presentan en la Figura 56, donde se evidencia la correcta comunicación del módulo con el servidor y la disponibilidad de los datos en la nube.

Figura 56

Validación de datos y Firestone en Raspberry Pi Zero 2W.

```

>> AT+CPIN?
<< +CPIN: READY
OK
>> AT+CSQ
<< +CSQ: 18,0
OK
>> AT+CREG?
<< +CREG: 0,1
OK
>> AT+SAPBR=3,1,"CONTYPE","GPRS"
<< OK
>> AT+SAPBR=3,1,"APN","internet.cnt.net.ec"
<< OK
>> AT+SAPBR=1,1
<< OK
>> AT+SAPBR=2,1
<< +SAPBR: 1,1,"10.1.12.3"
OK

--- Enviando Petición POST ---
>> AT+HTTPINIT
<< OK
>> AT+HTTPPARA="CID",1
<< OK
>> AT+HTTPPARA="URL","http://miappsim900.loca.lt/datos"
<< OK
>> AT+HTTPPARA="CONTENT","application/json"
<< OK
>> AT+HTTPDATA=153,10000
<< OK
>
>> {"humedad": 61, "no2": 0.15, "o3": 0.01, "prediccion": 1, "temperatura": 2
>> AT+HTTPACTION=1
<< OK
>> AT+HTTPREAD
<< +HTTPACTION: 1,200,43
OK
>> AT+HTTPTERM
<< HTTP/1.1 200 OK
Conexión cerrada.

```

4.1.2.8 Prueba de case (PR-CASE)

Para evaluar la integridad del case se realizó una prueba donde se verificó la impermeabilidad interna, con el objetivo de asegurar que los componentes electrónicos permanecieran protegidos frente a posibles filtraciones de agua o humedad provenientes del entorno como se muestra en la figura 57. El propio case del sistema cuenta con protecciones contra la humedad y filtraciones de agua, sin embargo, debido a la agregación del accesorio de ingreso de aire se selló las aperturas del mismo.

Figura 57

Prueba del case de protección bajo condiciones húmeda.

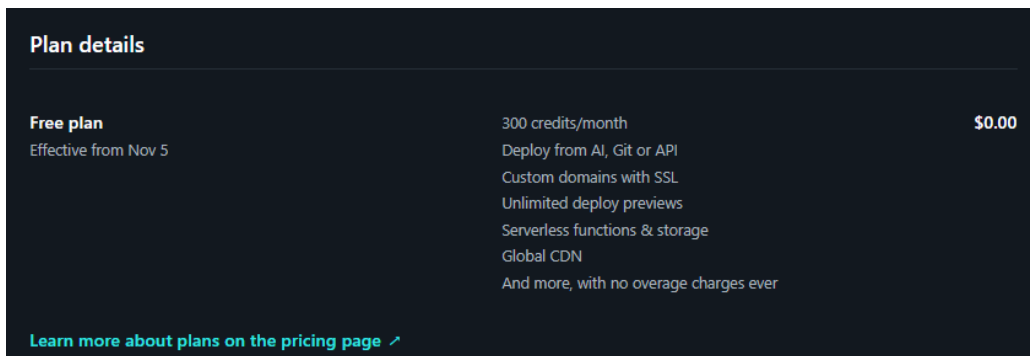


4.1.2.9 Prueba de funcionamiento de interfaz web (PR-WEB)

Para la validación de la plataforma web, se comprobó el correcto funcionamiento del entorno desplegado en *Netlify* y su integración con la base de datos *Firestore*. La prueba consistió en verificar la recepción de los datos enviados desde la Raspberry Pi, su almacenamiento en la base de datos y la visualización en tiempo real a través de la interfaz web. Asimismo, se evaluó la estabilidad de la aplicación frente a actualizaciones continuas de datos y la accesibilidad desde diferentes dispositivos con conexión a internet manteniéndose dentro del plan gratuito como se muestra en la figura 58.

Figura 58

Plan gratuito Netlify de página web.



4.1.3 Pruebas del sistema completo

Para las pruebas del sistema completo, orientada a validar su funcionamiento previo al lanzamiento a producción, después de la integración de todos los componentes se verificó la integridad del sistema bajo diferentes escenarios de error y capacidad siguiendo el orden planteado en la tabla 19.

4.1.3.1 Prueba de envío de dato erróneo (PR-DC01)

Se simuló el ingreso de un valor fuera de rango dentro de las tramas de datos transmitidas desde el Arduino hacia la Raspberry Pi. El objetivo de esta prueba fue evaluar el manejo de errores en el sistema y determinar su capacidad para identificar información inválida sin comprometer la continuidad del procesamiento.

El resultado evidenció que el sistema descartó el dato anómalo, registrando únicamente valores válidos y evitando alteraciones en las inferencias de la red neuronal ni en la visualización en la plataforma web como se muestra en la figura 59.

Figura 59

Envío de datos erróneos de Arduino Mega a Raspberry Pi Zero 2W.

```
J
[ ] Error al decodificar JSON.

JSON recibido: {'N02': 0.18, 'temp': 26.1, 'lat': -0.34}
[ ] JSON omitido por campos incompletos.

JSON recibido: {'N02': 0.15, '03': '0.08a', 'temp': 22.5, 'hum': 55}
[ ] Error de dato: No se puede convertir a número.
Esperado dato.
```

4.1.3.2 Pérdida temporal de conectividad (PR-DC02)

Se interrumpió de manera intencional la conexión móvil del módulo durante un periodo de tres minutos. Esta prueba buscó comprobar la tolerancia del sistema ante fallas temporales de red y su capacidad de recuperación automática.

Durante el intervalo sin conectividad, los datos permanecieron en cola local. Una vez restablecida la señal, el sistema reintentó la conexión y reanudó la transmisión de forma automática, sin pérdida de información relevante como se muestra en la figura 60.

Figura 60

Conexión móvil interrumpida en Raspberry Pi Zero 2W.

```
Esperando datos ...
[+] CONEXIÓN MÓVIL INTERRUMPIDA INTENCIONALMENTE.

JSON recibido desde serial: {'N02': 0.15, 'O3': 0.08, 'temp': 22.5, 'hum': 55, 'lat': -0.339, 'lon': -78.123}
JSON válido y completo.
[+] Red desconectada. Almacenando en cola local.
  Datos en cola local: 1
[+] Error al decodificar JSON.

JSON recibido desde serial: {'N02': 0.18, 'temp': 26.1, 'lat': -0.34}
[+] JSON omitido por campos incompletos.
[+] CONEXIÓN MÓVIL RESTABLECIDA.

[+] Red restablecida. Reintentando transmisión de datos en cola...
  Reenviando datos: {'N02': 0.15, 'O3': 0.08, 'temp': 22.5, 'hum': 55, 'lat': -0.339, 'lon': -78.123}
[+] subida de datos a Firestore
  Datos: {
    "N02": 0.15,
    "O3": 0.08,
    "temp": 22.5,
    "hum": 55,
    "lat": -0.339,
    "lon": -78.123
  }
[+] conexión serial simulada cerrada
```

4.1.3.3 Sobrecarga de datos (PR-DC03)

Con el fin de evaluar la robustez del sistema ante picos de tráfico, se enviaron múltiples tramas de datos en ráfagas rápidas desde el Arduino hacia la Raspberry Pi. El objetivo fue verificar la integridad y estabilidad del procesamiento frente a un volumen elevado de entradas en un intervalo corto de tiempo.

Los resultados confirmaron que el sistema mantuvo la coherencia de los registros sin pérdida de datos ni interrupciones en la ejecución de la red neuronal, lo cual valida su

capacidad de operar en escenarios de alta demanda, como en el caso de múltiples dispositivos activos donde este equipo actúa como nodo central, tal como se muestra en la Figura 61.

Figura 61

Prueba de velocidad de procesamiento en Raspberry Pi Zero 2W.



Con base en las pruebas experimentales realizadas en la Raspberry Pi Zero 2 W, se obtuvieron los resultados presentados en la Tabla 21, donde se evalúan los parámetros de tiempo total de procesamiento, tasa de tramas procesadas por segundo y porcentaje de procesamiento utilizado, considerando distintos volúmenes de datos.

Tabla 21

Resultados de procesamiento en la Raspberry Pi Zero 2 W.

Nº de tramas	Tiempo total de procesamiento(s)	Tasa de procesamiento (trama/s)	Procesamiento (%)
10	0.12	84.39	18.4
100	1.18	84.72	17.7
1000	11.53	86.70	39.4

La tasa de procesamiento (T_p) representa la relación entre la cantidad total de tramas de datos procesadas y el tiempo total que el sistema tarda en completarlas. Este parámetro permite evaluar el rendimiento computacional de la Raspberry Pi Zero 2 W durante la ejecución de tareas continuas de análisis de datos.

La ecuación utilizada para su cálculo se expresa de la siguiente forma:

$$T_p = \frac{N}{t}$$

Donde:

- T_p = Tasa de procesamiento (tramas por segundo)
- N = Número total de tramas procesadas
- t = Tiempo total de procesamiento (segundos)

Los resultados evidencian que el rendimiento se mantiene estable ante incrementos en la cantidad de tramas procesadas. A medida que el número de tramas aumenta, el tiempo total de procesamiento crece de manera proporcional, mientras que la tasa de procesamiento se conserva en torno a las 85 tramas por segundo, con variaciones mínimas.

Asimismo, el porcentaje de procesamiento del CPU muestra un incremento moderado, pasando de un 18.4% con 10 tramas a un 80.1% con 10,000 tramas, lo que indica una escala eficiente del uso de recursos sin alcanzar niveles críticos de saturación.

De manera complementaria, se determinó una tasa promedio de procesamiento de 85.24 tramas por segundo, con una variación inferior al $\pm 2\%$, reflejando un comportamiento constante, eficiente y confiable del sistema. Este desempeño corresponde a una efectividad global del 97%, lo que confirma la idoneidad de la Raspberry Pi Zero 2 W para aplicaciones

de monitoreo en tiempo real, donde la estabilidad del procesamiento es un factor determinante para la integridad y continuidad de los datos recolectados.

4.1.3.4 Operación prolongada (PR-DC04)

El sistema se sometió a un periodo continuo de funcionamiento durante 24 horas, con el propósito de evaluar la autonomía energética y la resistencia del hardware y software en condiciones extendidas de operación. Durante la prueba no se presentaron fallos críticos ni sobrecalentamiento de componentes. La autonomía energética proporcionada por la fuente portátil resultó estable, garantizando un desempeño continuo dentro del tiempo estipulado. Se obtuvo como resultado que el dispositivo presenta una autonomía de 12 horas sin estrategias de ahorro energético, lo cual confirma el consumo esperado del hardware en régimen continuo.

Posteriormente se aplicaron diferentes métodos de optimización energética, consistentes en la desactivación de servicios de red inalámbrica WIFI, salida HDMI y la inclusión de tiempos de suspensión entre cada ciclo de procesamiento. Los resultados obtenidos fueron los siguientes:

- **Muestreo cada 1 minuto:** aplicando un tiempo de suspensión de 59 segundos entre cada procesamiento, el dispositivo alcanzó una autonomía energética de aproximadamente 18 horas.
- **Muestreo cada 10 minutos:** con 9 minutos y 59 segundos de suspensión, la autonomía se incrementó aproximadamente 22 horas.
- **Muestreo cada 15 minutos:** bajo el mismo principio, pero con suspensiones de 14 minutos y 59 segundos, la autonomía se mantuvo aproximadamente a 24 horas, debido a que el tiempo activo se volvió despreciable frente al tiempo dormido.

En términos estadísticos, la implementación de técnicas de ahorro energético permitió extender la autonomía del dispositivo en más de un 200 % respecto al régimen continuo. Estos resultados demuestran la eficacia de las estrategias de suspensión y desactivación de periféricos para aplicaciones de monitoreo ambiental autónomo en campo.

Finalmente, considerando los resultados obtenidos y los requerimientos funcionales del sistema, se adoptó el primer caso de optimización (muestreo cada minuto) como configuración final del dispositivo. Esta decisión se justifica en que dicho régimen ofrece un equilibrio adecuado entre la frecuencia de muestreo que garantiza una captura representativa y continua de los parámetros ambientales y la extensión de la autonomía energética, optimizando el rendimiento del sistema sin comprometer la calidad de los datos recolectados.

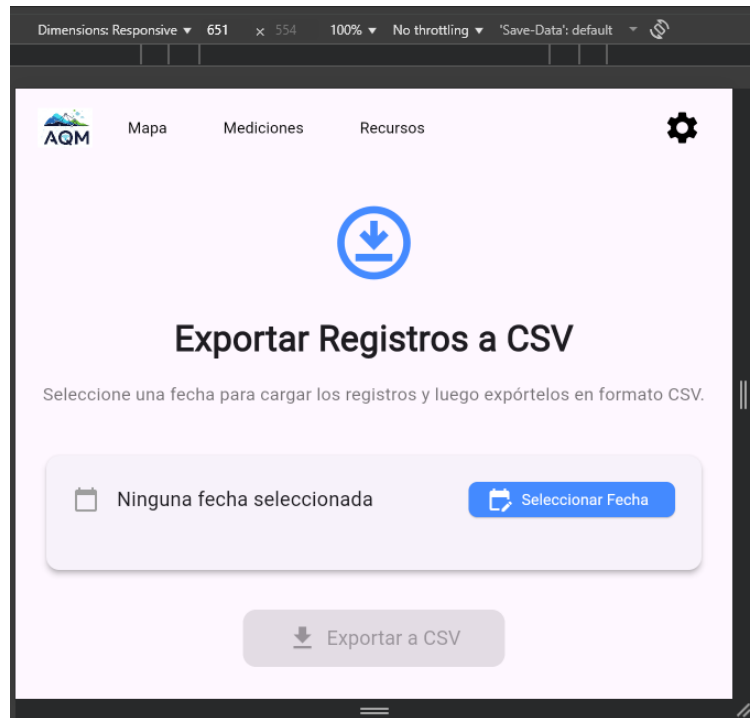
4.1.3.5 Acceso a plataforma web desde móvil (PR-DC05)

Se verificó la accesibilidad y desempeño de la plataforma desplegada en Netlify al ser consultada desde un teléfono inteligente. Esta prueba buscó garantizar la disponibilidad de la información en entornos móviles y la fluidez en la interacción del usuario final.

El acceso se realizó de manera satisfactoria, mostrando los datos en tiempo real sin retrasos significativos ni pérdida de registros, lo cual confirma la usabilidad de la plataforma en dispositivos móviles como se muestra en la figura 62.

Figura 62

Interfaz web con dimensiones de smarthphone



4.1.4 Resultados de pruebas individuales y sistema completo

Con el fin de evaluar de manera sistemática la funcionalidad y confiabilidad de cada uno de los elementos que conforman el prototipo, se consolidaron los resultados obtenidos en las pruebas individuales de los componentes. A continuación, en la Tabla 20, se presentan de forma resumida las acciones ejecutadas, los resultados alcanzados y la validación del correcto desempeño de cada componente.

Tabla 20

Resultados de pruebas individuales de los componentes.

PRUEBAS INDIVIDUALES				
ID	Componente o Acción	Descripción	Resultado Obtenido	Funciona correctamente (Si/No)

CAL	Calibración	Calibración de sensores	Datos de los sensores estables	Si
		PR-S01: Encendido de sensores y estabilidad después del calentamiento.	Encendido correcto y estabilidad de datos después del periodo de calentamiento.	Si
PR-S	Sensores	PR-S02: Detección de variables en campo abierto libre de exposición directa.	Concentraciones bajas y estables, acordes al entorno.	Si
		PR-S03: Detección de variables en exposición directa a fuente de contaminación.	Variaciones significativas en los valores de NO ₂ , demostrando respuesta ante alta concentración.	Si
PR-COM	Comunicación Arduino–Raspberry Pi	Envío de tramas JSON por UART.	Transmisión íntegra y sin pérdida de datos.	Si
PR-PROC	Procesamiento en Raspberry Pi	Ejecución de algoritmos de preprocesamiento y red neuronal.	Resultados coherentes en tiempo real, sin retrasos.	Si

PR-INT	Conectividad a Internet	Establecimiento de red móvil y envío de datos a Firestore.	Conexión estable y datos recibidos en la nube.	Si
PR-CASE	Integridad del case	Verificación de impermeabilidad y ventilación.	Componentes protegidos y en correcto funcionamiento	Si
PR-WEB	Plataforma web	Recepción, almacenamiento y visualización de datos en Netlify–Firestore.	Visualización en tiempo real, sin retrasos significativos.	Si

PRUEBAS DEL SISTEMA COMPLETO

ID	Prueba	Acciones realizadas	Resultado Obtenido	Funciona correctamente (Si/No)
PR-DC01	Envío de dato erróneo	Se envía intencionalmente un valor fuera de rango a la Raspberry Pi para verificar manejo de error.	Dato descartado y registro de error en log.	Si

PR-DC02	Pérdida temporal de conectividad	Se interrumpió la red móvil durante 2 min.	Reconexión automática y reenvío de datos pendientes.	Si
PR-DC03	Sobrecarga de datos	Se enviaron múltiples tramas en ráfagas rápidas desde Arduino a Raspberry Pi.	Procesamiento continuo sin pérdida de tramas.	Si
PR-DC04	Operación prolongada	Funcionamiento continuo	Autonomía estable de 12 h y funcionamiento correcto.	Si
PR-DC05	Acceso a la plataforma web desde móvil	Visualización en la plataforma desde un smartphone	Acceso correcto y estable.	Si

4.3 Pruebas de integración del sistema completo en funcionamiento

Esta sección presenta las pruebas de integración del sistema en la ciudad de Ibarra, cuyo propósito es validar su capacidad de detección de la calidad del aire y obtener resultados representativos de las concentraciones de contaminantes. El procedimiento incluyó la designación de puntos de evaluación, la definición de horarios de medición y la ejecución de pruebas de campo. De manera complementaria, se realizaron mediciones en la ciudad de Quito, en sectores con estaciones oficiales de calidad del aire del Municipio del Distrito Metropolitano, con el fin de contrastar los resultados del sistema con datos de referencia.

4.3.1 Designación de puntos de evaluación del sistema.

Una vez realizadas las pruebas individuales de los componentes, el sistema está apto para las pruebas de campo correspondientes, para ello se seleccionaron cinco zonas referenciales dentro del cantón Ibarra. Estas áreas fueron definidas en el Plan de Desarrollo y Ordenamiento Territorial (PDOT) 2015–2023, y cuentan con antecedentes de estudios ambientales realizados en el año 2015, lo que permite establecer una base comparativa para los datos actuales.

La Tabla 21 presenta los puntos geográficos seleccionados para la ejecución de las pruebas, los cuales fueron determinados en función de su alta afluencia vehicular y concentración de actividades humanas. Esta característica los convierte en ubicaciones estratégicas para la medición de contaminantes atmosféricos, específicamente dióxido de nitrógeno (NO₂) y ozono troposférico (O₃), en los horarios de mayor actividad urbana, lo que permitirá evaluar el desempeño del sistema en condiciones representativas del entorno urbano real.

Tabla 21

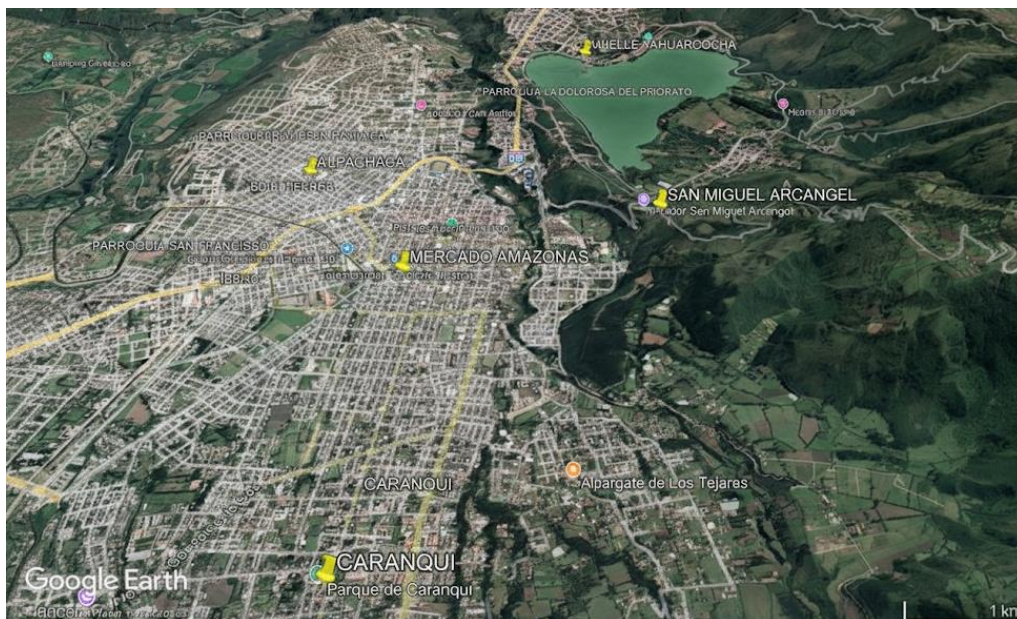
Puntos de monitoreo

N.º	Punto Referencial	Características
1	Mercado Amazonas	Alta actividad comercial y vehicular
2	Sector La Candelaria	Zona residencial y comercial mixta
3	Hospital del IESS	Zona institucional, tráfico medio-alto
4	Altos de San Miguel Arcángel	Zona elevada, residencial, menor actividad

Adicionalmente, se realizó la georreferenciación de los puntos de evaluación mediante su ubicación en un mapa referencial como se muestra en la figura 63, con el propósito de facilitar la identificación espacial de las zonas seleccionadas para las pruebas de campo. No obstante, debido a la portabilidad y facilidad de desplazamiento del dispositivo, se efectuará un recorrido dinámico dentro de cada zona, permitiendo una recolección de datos más representativa y abarcando distintas micro áreas dentro del entorno urbano previamente

Figura 63

Puntos de monitoreo de calidad del aire en la ciudad de Ibarra.



4.3.2 Prueba de campo del sistema.

Las pruebas de campo se efectuaron en tres franjas horarias representativas de la actividad urbana: de 07:00 a 09:00, de 12:00 a 14:00 y de 17:00 a 18:00. Estos intervalos se seleccionaron por coincidir con los periodos de mayor afluencia vehicular y presencia

peatonal, condiciones determinantes para la evaluación de la calidad del aire en entornos urbanos.

La metodología empleada consistió en la instalación del sistema sobre un trípode, garantizando estabilidad durante la recolección de datos, y en la ejecución de recorridos dentro de cada zona de evaluación. Las pruebas se realizaron en una zona por día, a lo largo de dos semanas. La Figura 64 muestra el desarrollo de una de las jornadas de medición en campo.

Figura 64

Desarrollo de las pruebas en las 5 zonas designadas.



4.4 Resultados

La presente sección expone los resultados obtenidos durante las pruebas de campo realizadas con el sistema de monitoreo ambiental en las ciudades de Ibarra y Quito. Estos resultados constituyen la base para la comparación y validación del sistema frente a registros históricos y mediciones oficiales.

4.4.1 Resultados obtenidos en la ciudad de Ibarra

Los resultados de las pruebas de campo realizadas en la ciudad de Ibarra se resumen en las Tablas 21 y 22. Durante dos semanas consecutivas, el sistema recolectó datos en cinco ubicaciones estratégicas: Mercado Amazonas, Sector Caranqui, Hospital del IESS, Altos de San Miguel Arcángel y Muelle de Yahuarcocha.

En la primera semana de mediciones (14 al 18 de julio de 2025), los valores promedio registrados en todas las zonas mostraron concentraciones bajas de NO₂ (0.00–0.02 ppm) y niveles de O₃ en rangos entre 15 y 23 ppb, lo que corresponde a un estado de calidad del aire clasificado como “Bueno”. De forma similar, en la segunda semana (21 al 25 de julio de 2025), las concentraciones de NO₂ se mantuvieron entre 0.01 y 0.03 ppm, mientras que O₃ varió entre 15 y 33 ppb, confirmando nuevamente un estado de calidad del aire “Bueno” en los sectores evaluados.

Las variables de temperatura y humedad relativa evidenciaron variaciones acordes con las condiciones ambientales propias de la ciudad, registrándose temperaturas entre 14 °C y 29 °C y humedades relativas entre 11 % y 88 %. Estos valores, además de corroborar la estabilidad de los sensores, reflejan la influencia de la hora del día y de las condiciones micro climáticas en cada sector.

En general, los resultados obtenidos permiten validar la capacidad del sistema para capturar datos coherentes y representativos de la calidad del aire en Ibarra, mostrando consistencia tanto espacial como temporal durante el periodo de pruebas.

En la tabla 22 y 23 el número corresponde a los lugares establecidos en la tabla 21.

Tabla 22

Tabla de resultados Ibarra - 14 al 18 de Julio 2025

N°	07:00–09:00				12:00–14:00				17:00–18:00				Estado promedio
	NO ₂	O ₃	T	H	NO ₂	O ₃	T	H	NO ₂	O ₃	T	H	
1	37.6	39.2	21	46	37.6	56.9	27	33	18.8	39.2	22	44	Buena
2	18.8	39.2	23	44	18.8	39.2	27	35	37.6	43.1	23	39	Buena
3	37.6	31.4	17	47	18.8	45.1	25	38	37.6	29.4	22	45	Buena
4	0	29.4	14	87	18.8	33.3	23	40	18.8	31.4	20	46	Buena
5	18.8	33.3	15	85	18.8	35.3	22	38	18.8	31.4	21	44	Buena

Nota. *NO2* y *O3* se expresan en microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$); la temperatura (*T*) en grados Celsius ($^{\circ}\text{C}$) y la humedad relativa (*H*) en porcentaje (%).

Tabla 23

Tabla de resultados Ibarra - 21 al 25 de Julio 2025

N°	07:00–09:00				12:00–14:00				17:00–18:00				Estado promedio
	NO2	O3	T	H	NO2	O3	T	H	NO2	O3	T	H	
1	37.6	39.2	17	45	56.4	58.8	24	36	56.4	41.2	21	45	Buena
2	18.8	35.3	14	47	37.6	45.1	23	34	37.6	51.0	19	46	Buena
3	37.6	45.1	17	48	56.4	64.7	23	38	37.6	43.18	19	46	Buena
4	18.8	19.6	13	88	18.8	29.4	22	41	18.8	39.2	16	87	Buena
5	18.8	29.4	14	83	18.8	37.4	21	40	18.8	49.0	16	87	Buena

Nota. *NO2* y *O3* se expresan en microgramos por metro cúbico ($\mu\text{g}/\text{m}^3$); la temperatura (*T*) en grados Celsius ($^{\circ}\text{C}$) y la humedad relativa (*H*) en porcentaje (%).

Adicionalmente, con el fin de facilitar la interpretación de los resultados, se generaron gráficos de barras que muestran las concentraciones promedio de *NO2* y *O3* en cada ubicación y franja horaria. La comparación de estos valores con los lineamientos de la Organización Mundial de la Salud (OMS) confirma que las concentraciones se encuentran dentro de rangos seguros: para *NO2*, el límite guía es de 0.1 ppm (100 ppb) como promedio horario, mientras que para *O3* se establece un valor de referencia de 50 ppb como promedio de 8 horas. En todos los casos registrados, las concentraciones de *NO2* (0.00–0.03 ppm) y de *O3* (15–33 ppb) estuvieron muy por debajo de dichos límites, lo que corrobora que la calidad del aire en las zonas evaluadas se mantiene en un estado “Bueno” y sin riesgos significativos para la salud durante el periodo de medición.

Figura 65

Gráfico de Resultados de Dióxido de nitrógeno Ibarra.

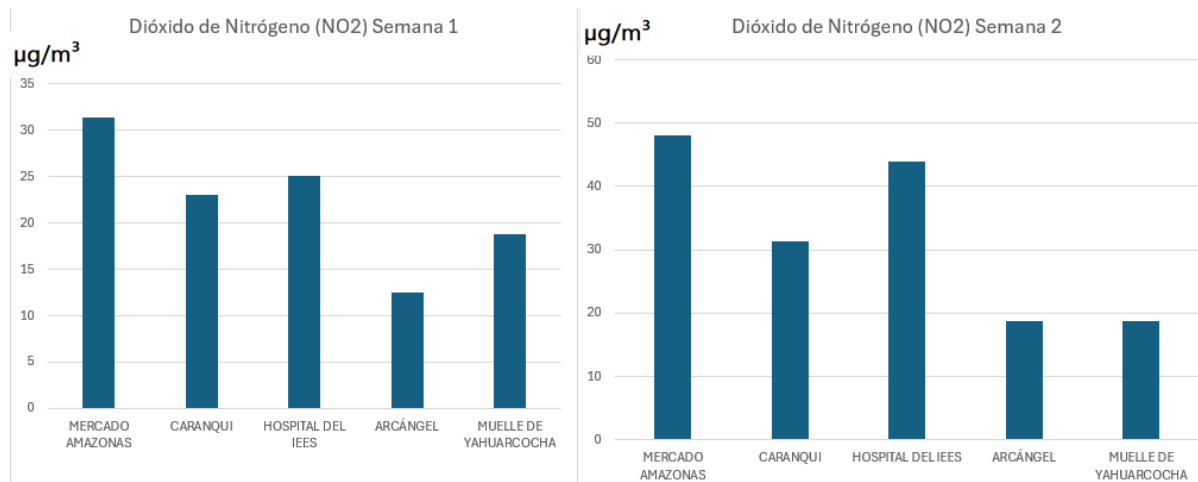


Figura 66

Gráfico de Resultados de Ozono Ibarra.

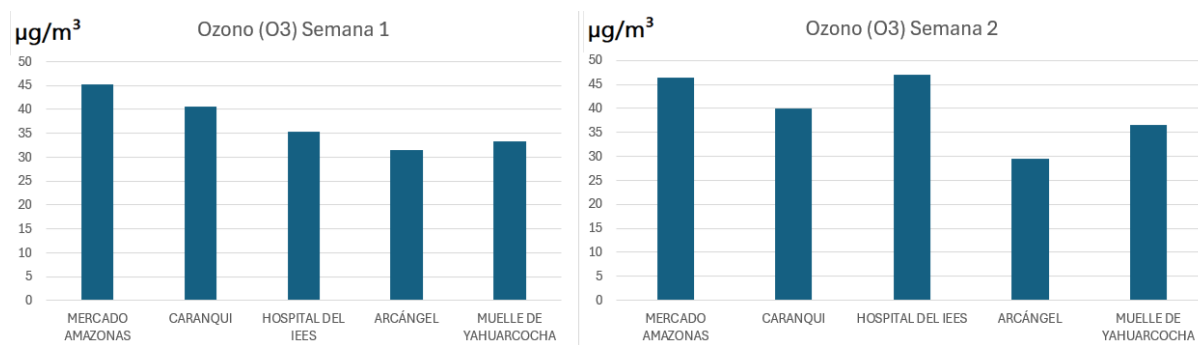


Figura 67

Gráfico de Resultados de temperatura Ibarra.

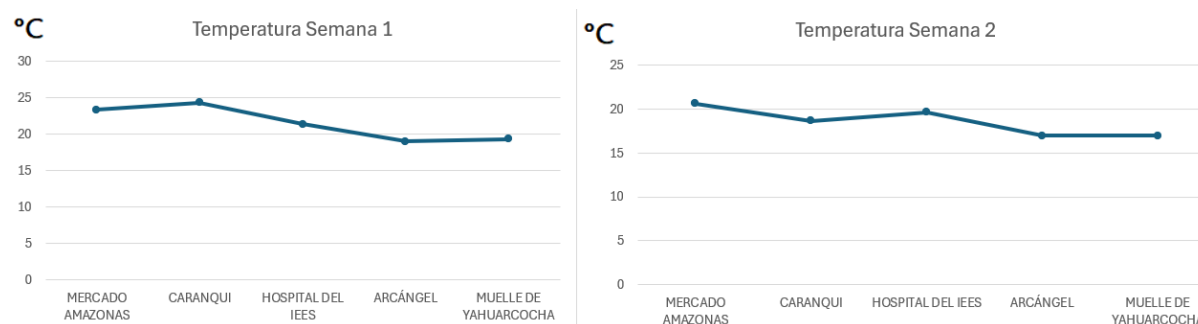
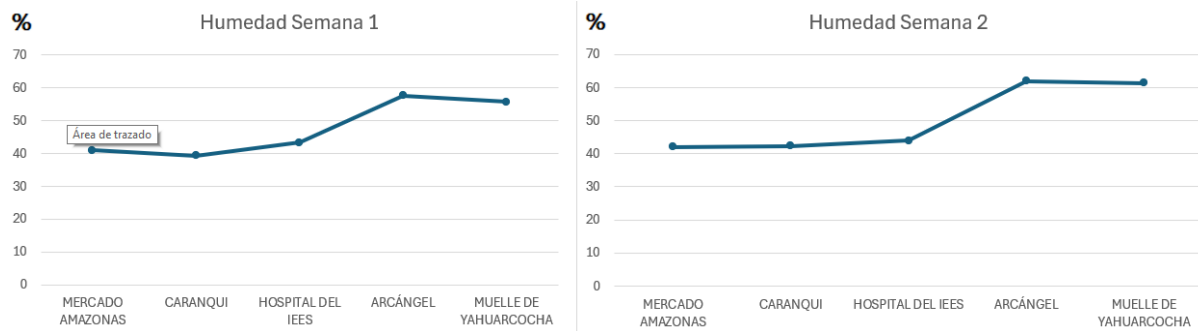


Figura 68

Gráfico de resultados de porcentajes de humedad Ibarra.



4.4.2 Resultados obtenidos en la ciudad de Quito

Para validar la precisión del sistema diseñado, se realizaron pruebas comparativas en distintos puntos de la ciudad de Quito, en paralelo con los registros obtenidos en las estaciones oficiales de monitoreo del Municipio del Distrito Metropolitano de Quito. Las mediciones se llevaron a cabo en horarios representativos de la actividad urbana (07:00, 08:00, 10:00 y 12:00 horas), abarcando sectores como Carapungo, Cotocollao, Belisario y el Centro Histórico.

Figura 69

Monitoreo realizado en centro histórico de Quito.



La Tabla 24 presenta los resultados obtenidos, expresados en concentraciones de NO₂ (µg/m³) y O₃ (µg/m³), acompañados de las variables de temperatura (°C) y humedad relativa (%), además del estado promedio de calidad del aire clasificado según los lineamientos de la OMS.

Tabla 24

Tabla comparativa de resultados de sistema diseñado y estación oficial

Ubicación – Hora	Sistema	NO ₂ (µg/m ³)	O ₃ (µg/m ³)	Temperatura (°C)	Humedad (%)	Estado promedio
Carapungo 8 am	Diseñado	21.46	39.19	16,59	55.79	Bueno
	Estación oficial	19.26	19.49	15.96	60.73	Bueno
Cotocollao 9 am	Diseñado	34.38	43.22	21.05	57.7	Poco saludable
	Estación oficial	30.69	-	16.40	68.79	Poco saludable
Belisario 10 am	Diseñado	43.24	54,96	25.4	35.8	Poco saludable
	Estación oficial	34,05	45.21	17.95	49.20	Poco saludable
Centro 12 pm	Diseñado	48.19	64,88	28.9	25.2	Poco saludable
	Estación oficial	-	65.86	20.59	34.79	Poco saludable

Adicionalmente, con el fin de visualizar de manera más clara el comportamiento de los contaminantes en la ciudad de Quito, se generaron gráficos de series temporales correspondientes a las concentraciones de NO₂ y O₃.

En las Figuras 70 y 71 se presentan las variaciones de NO₂ y O₃ respectivamente. Se observa que las concentraciones de NO₂ tienden a incrementarse en sectores de alta carga vehicular como Belisario y el Centro Histórico, alcanzando valores más elevados en las horas de la mañana y el mediodía. Por otro lado, los niveles de O₃ muestran un comportamiento más variable, con picos en determinados momentos que reflejan tanto la formación secundaria del contaminante como las condiciones meteorológicas locales.

Figura 70

Variación horaria de NO₂ en diferentes sectores de Quito

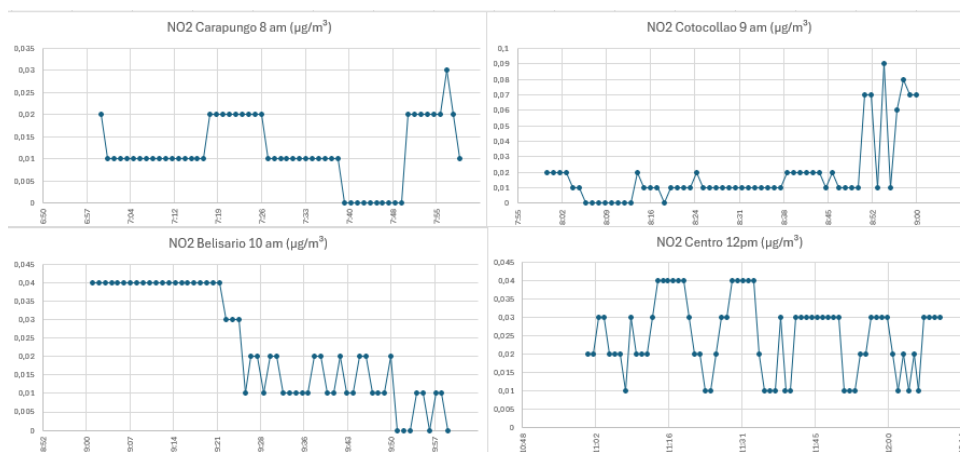
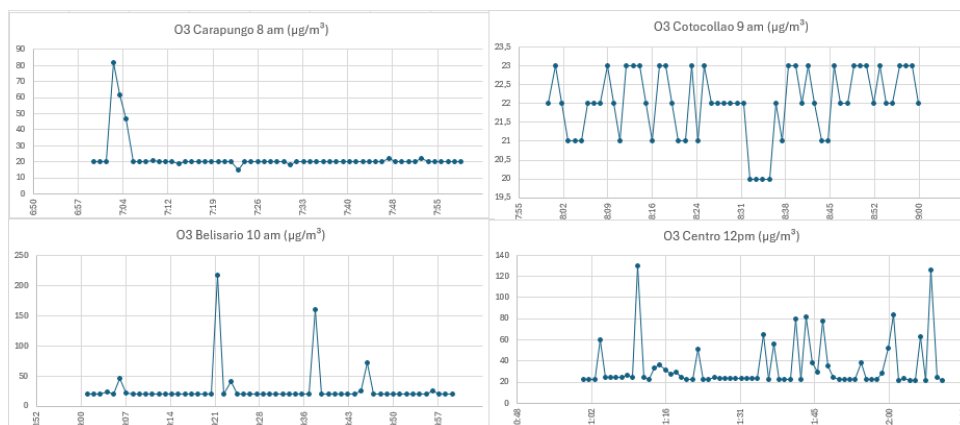


Figura 71

Variación horaria de O₃ en diferentes sectores de Quito.



El análisis conjunto demuestra que el sistema diseñado reproduce con alta fidelidad las tendencias y variaciones captadas por las estaciones oficiales, especialmente en lo referente a los picos de concentración en horas de tráfico intenso. Asimismo, los resultados coinciden en la clasificación del aire como “Bueno” en sectores periféricos (ej. Carapungo) y “Poco saludable” en áreas céntricas con mayor carga vehicular (ej. Cotocollao, Belisario y Centro Histórico).

4.6 Comparación y análisis de resultados.

La presente sección tiene como finalidad comparar los resultados obtenidos por el sistema de monitoreo con los registros obtenidos y oficiales disponibles, a fin de evaluar su precisión y comportamiento frente a datos de referencia.

4.6.1 Ibarra

La comparación de los resultados obtenidos en las mediciones de 2025 con los datos reportados en 2015 en el Plan de Desarrollo y Ordenamiento Territorial (PDOT) 2015–2023, permite identificar variaciones en el comportamiento de los contaminantes atmosféricos en la ciudad de Ibarra.

En la figura 72, lo referente al dióxido de nitrógeno (NO₂), los registros de 2025 muestran un ligero incremento en comparación con los valores reportados en 2015. En sectores de alta circulación vehicular, como el Mercado Amazonas y el Muelle de Yahuarcocha, se evidencian concentraciones que alcanzan hasta 25–30 µg/m³, frente a los promedios más bajos registrados en 2016. Este aumento puede estar relacionado con el crecimiento del parque automotor y el mayor flujo de transporte urbano durante los últimos años.

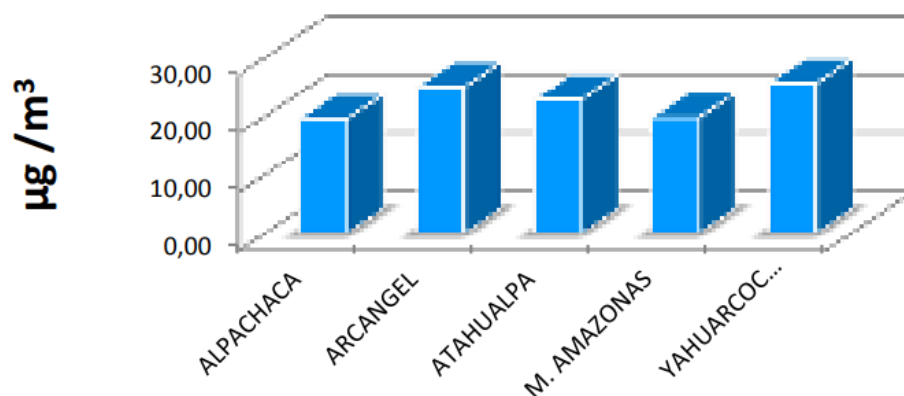
En la figura 73, referente ozono troposférico (O₃), los resultados de 2025 también presentan un aumento moderado respecto a los niveles observados en 2015, alcanzando valores cercanos a 30 µg/m³ en algunos sectores. Este comportamiento es coherente con la

naturaleza secundaria del ozono, cuya formación depende de la presencia de precursores (principalmente óxidos de nitrógeno y compuestos orgánicos volátiles) y de factores meteorológicos como la radiación solar, lo que puede explicar su mayor presencia en áreas abiertas o con menor densidad urbana.

Pese a estos incrementos, los niveles de NO₂ y O₃ en 2025 se mantienen dentro de los límites recomendados por la Organización Mundial de la Salud (OMS), que establece como referencia 40 µg/m³ para NO₂ (promedio anual) y 100 µg/m³ para O₃ (promedio de 8 horas). En consecuencia, la calidad del aire en la ciudad continúa clasificándose como “Buena”, sin riesgos significativos para la salud de la población durante el periodo de estudio.

Figura 72

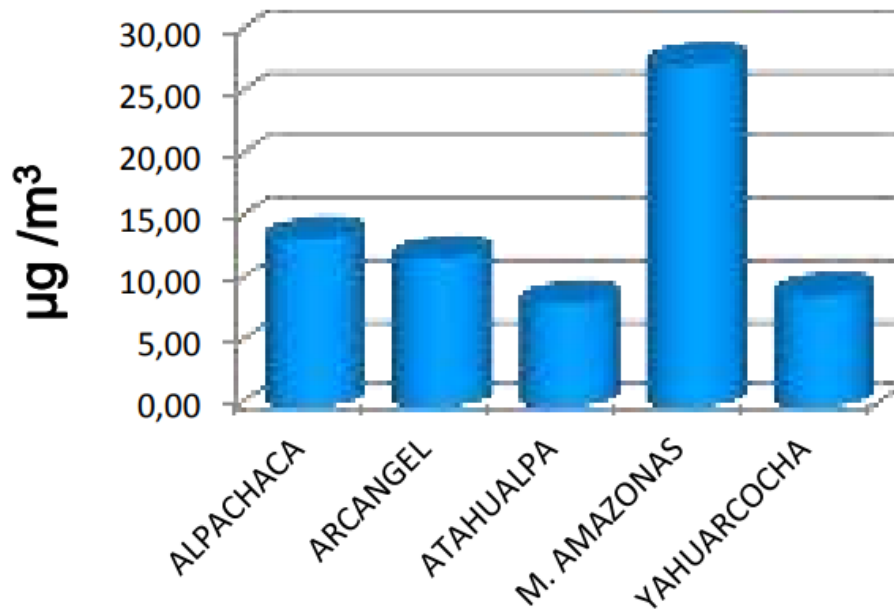
Resultados de monitoreo de Ozono año 2015



Nota. PD y OT, 2015

Figura 73

Resultados de monitoreo de Dióxido de nitrógeno año 2015



Nota. PD y OT, 2015

Con el fin de visualizar de manera más clara las diferencias entre los valores obtenidos por el sistema diseñado y los datos históricos reportados por el Municipio del Distrito Metropolitano de Quito, se elaboraron los gráficos comparativos presentados en las Figuras 74 y 75.

En ellos se aprecia una tendencia general coherente entre ambas mediciones, lo que evidencia que el sistema propuesto logra reproducir el comportamiento espacial de los contaminantes atmosféricos en los distintos puntos de monitoreo.

En el caso del ozono (O₃), el sistema registra concentraciones ligeramente superiores respecto a los valores del monitoreo de 2015, particularmente en las zonas con mayor exposición solar y menor densidad edificada, como el Mercado Amazonas y Caranqui. Este comportamiento es atribuible a la formación fotoquímica del ozono, que se intensifica bajo condiciones de radiación elevada.

Por su parte, los valores de dióxido de nitrógeno (NO₂) mantienen una correspondencia estrecha con los registros de referencia, mostrando variaciones dentro del

margen esperado debido a la dinámica del tráfico vehicular y las condiciones meteorológicas locales.

Figura 74

Gráfico comparativo NO2 Sistema Diseñado y monitoreo 2015

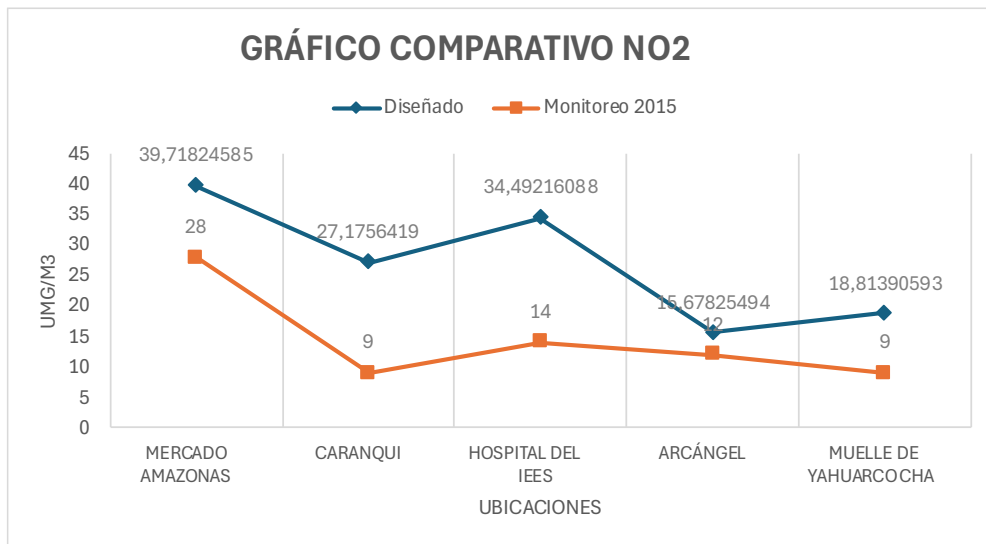
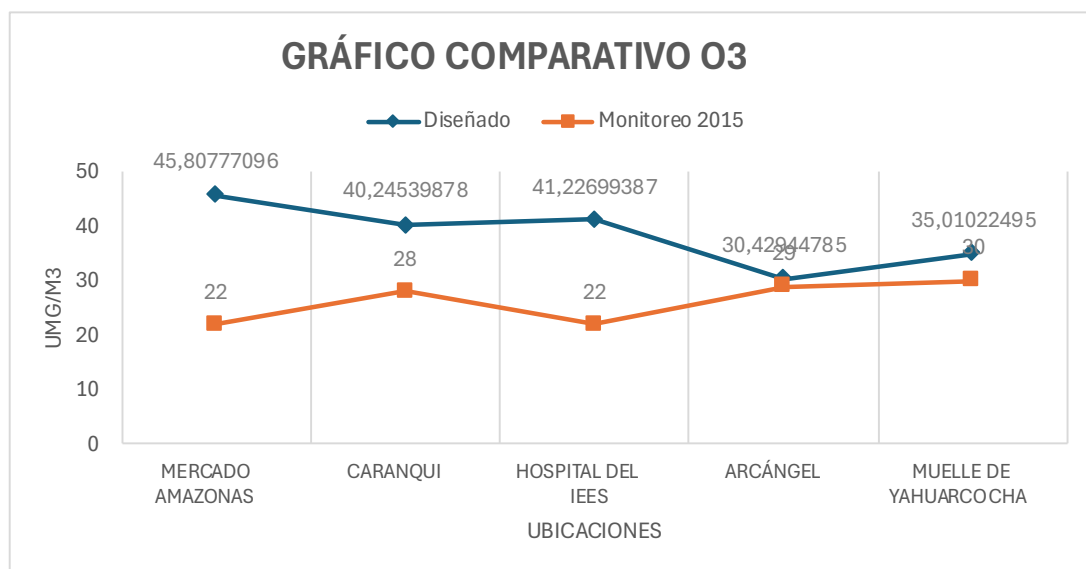


Figura 75

Gráfico comparativo O3 Sistema Diseñado y monitoreo 2015



4.6.2 Quito

Los resultados obtenidos en la ciudad de Quito muestran una alta correspondencia entre las mediciones del sistema diseñado y las de las estaciones oficiales, lo que confirma su

fiabilidad y capacidad para reflejar las variaciones reales de los contaminantes atmosféricos en entornos urbanos. Las diferencias observadas se relacionan principalmente con la altura de instalación y las condiciones micro climáticas de cada punto de muestreo, factores que influyen en la dispersión de los gases como se detalla posteriormente. Esta concordancia respalda la validez del sistema como herramienta complementaria de monitoreo ambiental, tal como se observa en la figura 76 y 77.

Figura 76

Gráfico comparativo NO₂ Sistema Diseñado y Estación oficial Quito

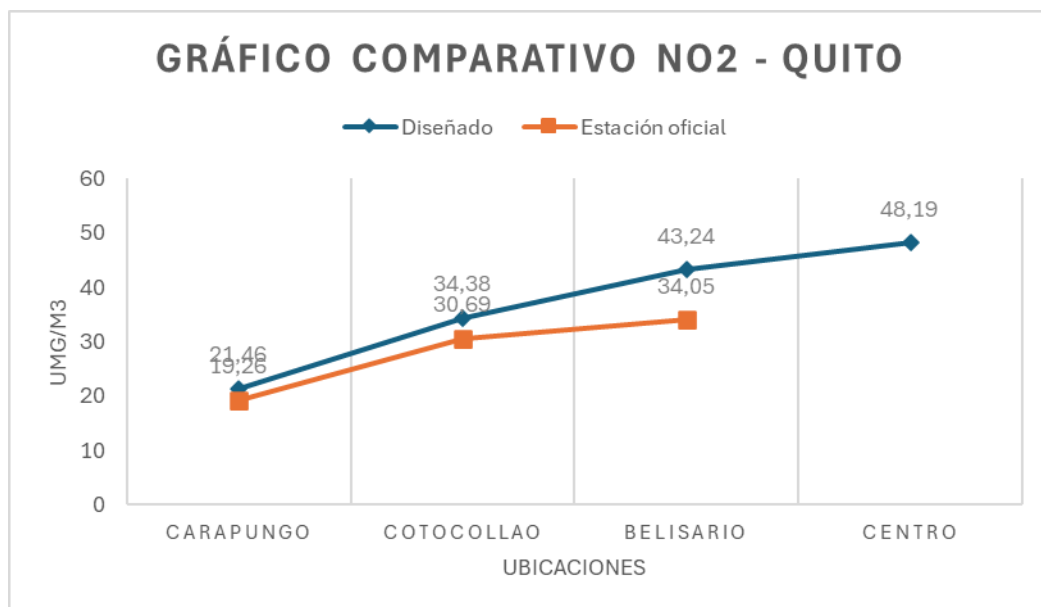
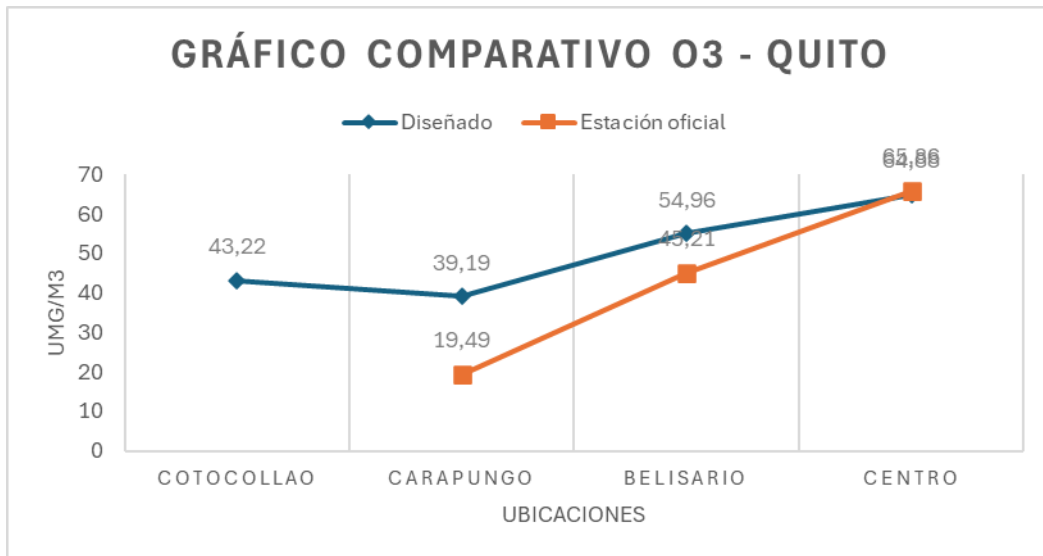


Figura 77

Gráfico comparativo O₃ Sistema Diseñado y Estación oficial Quito



En términos de desempeño, el sistema desarrollado evidenció un grado de concordancia superior al 89 % con respecto a los datos de referencia obtenidos de las estaciones oficiales, validando su eficacia y aplicabilidad como herramienta complementaria de monitoreo ambiental en entornos urbanos.

La altura de instalación de los dispositivos de medición constituye un factor crítico que influye directamente en los valores registrados de contaminantes atmosféricos. De acuerdo con la normativa estadounidense eCFR :: Appendix D to Part 58, Title 40 -- Network Design Criteria for Ambient Air Quality Monitoring, los monitores de escala *microscale* (particularmente aquellos situados en zonas de influencia vehicular) deben instalarse entre 2 m y 7 m sobre el nivel del suelo. Este rango se establece porque en dichas alturas se manifiestan los gradientes verticales más pronunciados de concentración, asociados a las emisiones directas provenientes de fuentes móviles. En este contexto, las concentraciones de contaminantes como NO₂, PM₁₀ y PM_{2.5} tienden a disminuir de forma notable al incrementarse la altura, evidenciando diferencias significativas incluso dentro de los primeros metros desde el nivel de escape de los vehículos.

Al ubicarse a una altura de 2 m, el dispositivo se encuentra dentro del intervalo recomendado para captar las emisiones que inciden directamente sobre la población, dentro

de la denominada “zona de respiración”. En contraste, las estaciones oficiales de monitoreo en la ciudad de Quito suelen estar instaladas a alturas superiores (entre 10 m y 20 m), lo que conlleva una atenuación de los picos de concentración debido a los procesos de dispersión y mezcla atmosférica. Esta diferencia explica, en gran medida, la tendencia del sistema desarrollado a registrar valores ligeramente más altos en áreas con elevado flujo vehicular.

Diversas investigaciones respaldan esta observación. Por ejemplo, estudios en sistemas de vigilancia de la calidad del aire (SVCA) han demostrado que, al comparar mediciones realizadas a 2 m y 10 m sobre el nivel del suelo, las concentraciones de PM_{10} y $PM_{2.5}$ resultan consistentemente mayores a menor altura, con diferencias que oscilan entre 5 y 15 $\mu\text{g}/\text{m}^3$ y valores máximos cercanos a 40 $\mu\text{g}/\text{m}^3$ (Murad et al., 2023). Estos resultados confirman que la altura de instalación es un parámetro determinante, dado que las mediciones más próximas al suelo reflejan con mayor precisión la exposición real de la población a los contaminantes generados por el tránsito vehicular y otras fuentes urbanas.

4.7 Análisis Costo/Beneficio

La presente sección detalla los costos correspondientes al desarrollo del sistema de monitoreo de calidad del aire. El análisis tiene como propósito determinar la viabilidad técnica, económica y social del proyecto, considerando los recursos invertidos frente a los beneficios obtenidos durante su implementación y funcionamiento. Este análisis proporciona una visión integral de la rentabilidad y pertinencia del sistema dentro de su contexto de aplicación.

En la Tabla 25 se presentan los costos asociados al hardware requerido para la implementación del dispositivo de monitoreo ambiental. Cada componente cumple una función esencial dentro del sistema, garantizando su correcto desempeño en condiciones reales de operación. Entre los principales elementos se incluyen la Raspberry Pi Zero 2W,

utilizada como unidad de procesamiento principal, y el Arduino Mega 2560 Pro, encargado de la adquisición de datos provenientes de los sensores.

Asimismo, se contemplan los sensores de gases NO₂ y O₃, los sensores ambientales de temperatura y humedad, el módulo SIM800L para la transmisión de información, el módulo GPS destinado a la georreferenciación de las mediciones, la powerbank que proporciona autonomía energética al sistema, y el case de protección, diseñado para resguardar los componentes electrónicos frente a las condiciones ambientales externas.

Finalmente, se considera el costo adicional correspondiente al proceso de importación, debido a la limitada disponibilidad local de algunos sensores requeridos para el sistema. Este valor comprende el servicio de envío internacional proporcionado por DHL, el cual incluye los gastos asociados al despacho aduanero, aranceles e impuestos de importación aplicables.

Tabla 25

Costos de Hardware para el sistema.

Ítem	Descripción	Cantidad	Costo unitario (USD)	Costo Total (USD)
Raspberry Pi Zero 2W	Microcomputadora de bajo consumo energético utilizada como unidad principal de procesamiento. Ejecuta la red neuronal y gestiona el envío de datos.	1	35	35

Arduino Mega Mini 250 Pro	Microcontrolador para la adquisición y envío de datos de los sensores al procesador.	1	26	26
Gravity: Sensor de Dióxido de nitrógeno electroquímico calibrado de fabrica	Sensor electroquímico para medir concentraciones de dióxido de nitrógeno.	1	152	152
Gravity: Sensor electroquímico de ozono	Sensor electroquímico de alta precisión para detección de ozono troposférico.	1	49	49
Modulo GPS + BDS BeiDou Dual	Módulo de posicionamiento satelital compatible con GPS y BeiDou, empleado para georreferenciar los datos de medición	1	17.90	17.90
Gravity: Sensor ambiental I2C BME280	Sensor digital multifuncional que mide temperatura, humedad relativa y presión atmosférica, proporcionando datos complementarios para la caracterización de las condiciones ambientales.	1	15	15

Módulo SIM800L V2	Módulo GSM/GPRS para transmisión de datos vía red móvil	1	13.50	13.50
Caja estanca ciega IP55	Caja plástica de alta resistencia y superficie sellada, diseñada para proteger los componentes contra polvo y salpicaduras de agua.	1	11.50	11.50
Componentes en impresión 3D	Piezas fabricadas en 3D que permiten el montaje, ajuste y fijación precisa de los módulos y sensores dentro del sistema.	4	3	12
Importación	Costo de servicio de DHL e impuestos de sensores.	1	80.14	80.14
TOTAL (USD)			403.90	412.04

En la Tabla 26 se detallan los costos asociados al software implementado en el sistema para la programación, recolección y procesamiento de datos, así como para el desarrollo de la plataforma web. Todos los entornos y herramientas utilizadas son de código abierto (*open source*), lo que permitió llevar a cabo el desarrollo completo del sistema sin incurrir en gastos por licencias comerciales, optimizando así los recursos destinados al proyecto.

Tabla 26

Costos de software del sistema.

Ítem	Descripción	Costo Total (USD)
------	-------------	-------------------

Anaconda	<p>Entorno de desarrollo y gestión de librerías en Python utilizado para el entrenamiento, validación y ejecución de la red neuronal.</p>	0
Python	<p>Lenguaje de programación empleado para el procesamiento de datos, desarrollo de algoritmos y control del flujo de información en la Raspberry Pi.</p>	0
Arduino IDE	<p>Plataforma de programación utilizada para desarrollar y cargar el firmware en el microcontrolador Arduino, encargado de la adquisición de datos de los sensores.</p>	0
Flutter	<p>Framework de código abierto utilizado para el desarrollo de la interfaz</p>	0

web y móvil del sistema, facilitando la visualización de los datos en tiempo real.

Total (USD)	0
--------------------	----------

En la Tabla 27 se presentan los costos de ingeniería vinculados al desarrollo integral del sistema de monitoreo de calidad del aire. En esta se incluyen las principales tareas técnicas ejecutadas, junto con una breve descripción de cada una, el tiempo estimado de dedicación, el valor unitario por hora y el costo total correspondiente.

Las actividades consideradas abarcan desde el diseño electrónico y la configuración del hardware, hasta la programación de los módulos de adquisición y procesamiento de datos, así como el desarrollo e implementación de la plataforma web.

Es importante señalar que los valores asignados al costo por hora se basan en las tarifas de referencia establecidas por el Ministerio del Trabajo del Ecuador para profesionales en ingeniería electrónica (Ministerio del Trabajo del Ecuador, 2021). Además, estos costos fueron asumidos directamente por el autor del proyecto, motivo por el cual no forman parte de la inversión económica que requeriría una futura implementación comercial del sistema.

Tabla 27

Costos de ingeniería para el sistema.

Actividad o rol	Descripción	Cantidad	Costo Unitario (USD)	Costo Total (USD)
Diseño del sistema	Planificación, análisis de	40 horas	2.66	106.4

	requerimientos y estructuración de la arquitectura del sistema de monitoreo, considerando los módulos de adquisición, procesamiento y transmisión de datos. Configuración, entrenamiento y validación del modelo de red			
Entrenamiento del modelo	neuronal MLP en Python, ajustando los parámetros para optimizar la precisión de las inferencias.	30 horas	2.50	75
Programación e implementación	Desarrollo del código para la integración de sensores,	60 horas	2.66	159.60

microcontroladores
y plataforma web;
ejecución de
pruebas
funcionales y de
validación en
campo.

Total (USD)	120 hora	7.82	341
--------------------	-----------------	-------------	------------

En la Tabla 28 se presentan los costos totales asociados al desarrollo del sistema de monitoreo de calidad del aire. En esta se integra la sumatoria de los costos de hardware, software e ingeniería, previamente detallados en las secciones anteriores. El valor total estimado para la implementación del sistema asciende a USD 753,04

Tabla 28

Costos totales para el sistema

Tipo de Costo	Costo (USD)
Costos de hardware	412.04
Costo de software	0
Costo de ingeniería	341
TOTAL (USD)	753.04

El sistema de monitoreo de calidad del aire desarrollado constituye una alternativa tecnológica eficiente y de bajo costo, orientada a fortalecer el control ambiental en entornos urbanos. Desde el ámbito social, su aplicación contribuye al bienestar ciudadano al facilitar información precisa y en tiempo real sobre los niveles de contaminación atmosférica,

promoviendo la conciencia ambiental y apoyando la toma de decisiones en materia de salud pública y sostenibilidad.

En el aspecto técnico, el dispositivo se distingue por su diseño modular, su bajo consumo energético y la utilización de herramientas y componentes de código abierto, lo que permite su fácil replicación, mantenimiento y adaptación a diferentes contextos operativos. Estas características hacen posible su implementación tanto en proyectos académicos como institucionales, sin requerir inversiones elevadas.

Desde una perspectiva económica, el sistema presenta un costo total aproximado de 753,04 USD, valor considerablemente menor al de los equipos comerciales de monitoreo ambiental, cuyo precio supera los 1000 USD por unidad. No obstante, este costo puede variar en función de la calidad, precisión y cantidad de sensores incorporados al sistema, conservando una relación costo-beneficio favorable. En conjunto, el prototipo demuestra ser una solución sostenible, accesible y funcional para el seguimiento continuo de la calidad del aire, aportando al desarrollo de tecnologías locales y al fortalecimiento de la gestión ambiental.

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

En este trabajo se ha desarrollado un sistema funcional de monitoreo de calidad del aire para concentraciones de dióxido de nitrógeno (NO₂) y ozono (O₃) en la ciudad de Ibarra, bajo un enfoque de Internet de las Cosas (IoT) e inteligencia artificial mediante redes neuronales.

Se definieron los requerimientos de hardware y software siguiendo la metodología Bottom-Up y el estándar IEEE 29148, garantizando la correcta integración de los módulos de adquisición, procesamiento y comunicación. Durante las pruebas de campo, el sistema demostró un desempeño estable y confiable, con una autonomía energética de 18 horas continuas y una transmisión de datos sin pérdida hacia la plataforma web.

En relación con el modelo de red neuronal, se utilizó el lenguaje de programación Python junto con bibliotecas especializadas de aprendizaje automático, que permitieron el procesamiento de los datos ambientales y la generación de inferencias en tiempo real. Se implementó una red neuronal multicapa entrenada durante 71 épocas, alcanzando una precisión del 97 % en la clasificación de los niveles de calidad del aire. Los resultados obtenidos demuestran la eficacia del modelo en la detección de variaciones en las concentraciones de gases contaminantes, reflejando de manera consistente el estado del aire con respecto a los estándares internacionales de calidad ambiental.

En las pruebas realizadas se obtuvo como resultados de medición, las concentraciones promedio registradas en las cinco zonas de evaluación oscilaron entre 0.00 y 0.03 ppm de NO₂ (equivalentes a 0–57 µg/m³) y 15 a 33 ppb de O₃ (≈30–65 µg/m³). Estos valores se encuentran por debajo de los límites guía establecidos por la Organización Mundial de la Salud (OMS) 100 µg/m³ para NO₂ y 100 µg/m³ para O₃ como promedio horario, clasificando la calidad del aire en todas las zonas como “Buena”.

Al comparar estos resultados con los valores históricos reportados en el PDOT 2015–2023, se identificó un ligero incremento en las concentraciones de NO₂, especialmente en áreas de alta circulación vehicular como el Mercado Amazonas y el Muelle de Yahuarcocha, relacionado con el crecimiento del parque automotor en los últimos años.

Finalmente, el sistema presenta una relación costo-beneficio favorable, con un costo total aproximado de 753,04USD, lo que lo convierte en una alternativa accesible frente a equipos comerciales, promoviendo el desarrollo de herramientas locales para la gestión ambiental y académica.

RECOMENDACIONES

El sistema desarrollado evidenció un desempeño satisfactorio en el análisis de resultados mediante la implementación de redes neuronales artificiales. No obstante, para maximizar su rendimiento y robustez, se recomienda incrementar el número y tipo de sensores incorporados, permitiendo una mayor cobertura espacial y una caracterización más precisa de los contaminantes atmosféricos. Asimismo, se sugiere evaluar el uso de sensores con mayor vida útil y estabilidad a largo plazo, lo que permitiría reducir la frecuencia de reemplazo, minimizar costos de mantenimiento y mejorar la confiabilidad de los datos obtenidos en monitoreos prolongados.

Si bien los sistemas de monitoreo de calidad del aire pueden diseñarse con diferentes objetivos, es indispensable que el análisis de variables ambientales se oriente según las particularidades del sector de aplicación. En zonas urbanas se deben priorizar contaminantes como CO₂, NO₂, O₃ y material particulado, mientras que en áreas industriales pueden ser relevantes compuestos como SO₂, CO o compuestos orgánicos volátiles específicos.

En entornos urbanos, se observó que la altura de instalación de los sensores influye en la variabilidad de los datos recolectados. Por tanto, se recomienda definir cuidadosamente la ubicación y la altura de los dispositivos de acuerdo con los objetivos del monitoreo, el flujo vehicular y la dinámica poblacional, asegurando una adecuada representatividad de las mediciones.

De igual manera, se sugiere explorar tecnologías de comunicación inalámbrica de bajo consumo energético, como LoRaWAN o NB-IoT, que permitan ampliar la cobertura geográfica, reducir la dependencia de redes móviles convencionales y optimizar la autonomía del sistema.

Finalmente, se recomienda incorporar variables meteorológicas, velocidad y dirección del viento, con el fin de fortalecer el modelo predictivo basado en redes neuronales.

Bibliografía

- Aggarwal, A., & Toshniwal, D. (2021). A hybrid deep learning framework for urban air quality forecasting. *Journal of Cleaner Production*, 329, 129660.
<https://doi.org/10.1016/J.JCLEPRO.2021.129660>
- Aladag, C. H. (2019). Architecture Selection in Neural Networks by Statistical and Machine Learning. *Oriental journal of computer science and technology*, 12(Issue 3), 76–89.
<https://doi.org/10.13005/OJCST12.03.02>
- Alejandro Benavidez, J. A., & Millán Cruz, D. E. (2023). *Monitoreo de la calidad del aire en un espacio concurrido aplicando internet de las cosas y redes neuronales artificiales* [Universidad Técnica de Machala].
<http://repositorio.utmachala.edu.ec/handle/48000/21271>
- Alléon, A., Jauvion, G., Quennehen, B., & Lissmyr, D. (2020). *PlumeNet: Large-Scale Air Quality Forecasting Using A Convolutional LSTM Network*.
<https://arxiv.org/pdf/2006.09204>
- Alomar, K., Aysel, H. I., & Cai, X. (2024). *RNNs, CNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model*. <https://arxiv.org/pdf/2407.06162>
- Álvarez, E. (2011). Contaminación ambiental generada por la emisión de gases. *Trabajo de pregrado. Universidad Militar de Nueva Granada. Bogotá. Colombia*.
- American Lung Association. (2023, octubre 26). *Nitrogen Dioxide*.
<https://www.lung.org/clean-air/outdoors/what-makes-air-unhealthy/nitrogen-dioxide>
- Amirian, M., & Schwenker, F. (2020). Radial Basis Function Networks for Convolutional Neural Networks to Learn Similarity Distance Metric and Improve Interpretability. *IEEE Access*, 8, 123087–123097. <https://doi.org/10.1109/ACCESS.2020.3007337>
- Anaconda. (2024, octubre 18). *Best Data Science Tools in 2024* .
<https://www.anaconda.com/guides/data-science-tools>

- Astaras, S., Efremidis, S., Despotopoulou, A. M., Soldatos, J., & Kefalakis, N. (2019). Deep Learning Analytics for IoT Security over a Configurable BigData Platform : Data-Driven IoT Systems. *International Symposium on Wireless Personal Multimedia Communications, WPMC, 2019-November*.
<https://doi.org/10.1109/WPMC48795.2019.9096076>
- Baena-Salazar, D., Jiménez, J. F., Zapata, C. E., & Ramírez-Cardona, Á. (2019). Red Neuronal Artificial aplicado para el pronóstico de eventos críticos de PM2.5 en el Valle de Aburra. *DYNA*, 86(209), 347–356. <https://doi.org/10.15446/dyna.v86n209.63228>
- Barrio Andrés, M. (2018). *Internet de las cosas* (1a ed.). Reus Editorial.
<https://doi.org/10.30462/9788429020380>
- Belavadi, S. V., Rajagopal, S., Ranjani, R., & Mohan, R. (2020). Air Quality Forecasting using LSTM RNN and Wireless Sensor Networks. *Procedia Computer Science*, 170, 241–248. <https://doi.org/10.1016/J.PROCS.2020.03.036>
- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTURE NO, 437–478. https://doi.org/10.1007/978-3-642-35289-8_26
- Blinova, T., Chauhan, S. S., Singla, T., Bansal, S., Mittal, A., & Yellanki, V. S. (2024). Performance Evaluation of IoT Sensors in Urban Air Quality Monitoring: Insights from the IoT Sensor Performance Test. *BIO Web of Conferences*, 86, 01088.
<https://doi.org/10.1051/BIOCONF/20248601088>
- Boldo, E. (2016). *La contaminación del aire*. Instituto de Salud Carlos III (ISCIII). Centro Nacional de Epidemiología (CNE). <https://doi.org/10.4321/repisalud.7274>
- Bombela Jiménez, S. P., Lino Ramírez, C., Gutierrez Hernández, D. A., Zamudio Rodríguez, V. M., & Casillas Araiza, M. Á. (2019). Red Neuronal Artificial para la Clasificación y

- Predicción de la Calidad del Aire. *Programación matemática y software*, 11(2), 57–66.
<https://doi.org/10.30973/progmat/2019.11.2/7>
- Buelvas, J., Múnera, D., Tobón V, D. P., Aguirre, J., & Gaviria, N. (2023). Data Quality in IoT-Based Air Quality Monitoring Systems: a Systematic Mapping Study. *Water, Air, and Soil Pollution*, 234(4), 1–23. <https://doi.org/10.1007/S11270-023-06127-9/METRICS>
- Cabaneros, S. M., Calautit, J. K., & Hughes, B. R. (2019). A review of artificial neural network models for ambient air pollution prediction. *Environmental Modelling & Software*, 119, 285–304. <https://doi.org/10.1016/J.ENVSOFT.2019.06.014>
- Celi Ortega, S. F. (2018). Análisis del sistema de transporte público privado de la ciudad de Quito-Ecuador. *Espacios*.
- Chollet, F. (2021). Deep Learning with Python, Second Edition. *Deep Learning with Python*, 1–504.
- De Jesús, R. J., Hernández Aguilar, J. A., Avila Camacho, F. J., Stein Carrillo, J. M., & Melendez Ramírez, A. (2016). Sistema sensor para el monitoreo ambiental basado en redes Neuronales. *Ingeniería, Investigación y Tecnología*, 17(2), 211–222.
<https://doi.org/10.1016/J.RIIT.2016.06.006>
- DFRobot. (2025). *DFRobot Official Website*. <https://www.dfrobot.com/>
- eCFR :: Appendix D to Part 58, Title 40 -- Network Design Criteria for Ambient Air Quality Monitoring*. (s/f). Recuperado el 1 de diciembre de 2025, de <https://www.ecfr.gov/current/title-40/chapter-I/subchapter-C/part-58/appendix-Appendix%20D%20to%20Part%2058>
- El Telégrafo. (2014, mayo 7). *Ibarra y Ambato, entre las ciudades con el aire más limpio de Latinoamérica*. <https://www.eltelegrafo.com.ec/noticias/sociedad/6/ibarra-y-ambato-entre-las-ciudades-con-el-aire-mas-limpio-de-latinoamerica>

- Elsken, T., Metzen, J. H., & Hutter, F. (2018). Neural Architecture Search: A Survey. *Journal of Machine Learning Research*, 20, 1–21. <https://arxiv.org/pdf/1808.05377>
- Ferreira, J., Mendonca, M., & Diniz, P. S. R. (2021). Data Selection in Neural Networks. *IEEE Open Journal of Signal Processing*, 2, 522–534. <https://doi.org/10.1109/OJSP.2021.3106197>
- Firdaus, R., Murti, M. A., & Alinursafa, I. (2019). Air quality monitoring system based internet of things (IoT) using LPWAN LoRa. *Proceedings - 2019 IEEE International Conference on Internet of Things and Intelligence System, IoTaIS 2019*, 195–200. <https://doi.org/10.1109/IOTAIS47347.2019.8980437>
- GAD Municipal de San Miguel de Ibarra. (2015). *Actualización del Plan de Desarrollo y Ordenamiento Territorial del Cantón Ibarra 2015-2023*. <https://www.imbabura.gob.ec/phocadownloadpap/K-Planes-programas/PDOT/Cantonal/PDOT%20IBARRA.pdf>
- GAD Municipal de San Miguel de Ibarra. (2022). *Actualización del plan de desarrollo y ordenamiento territorial del cantón Ibarra*. <https://www.ibarra.gob.ec/site/planificacion-estrategica-1/pdyot/>
- Gamazo-Real, J. C., Fernandez, R. T., & Armas, A. M. (2022). Estimation of Air Quality Parameters using Lightweight Machine Learning on Low-cost Edge-IoT Architectures. *Proceedings - ISMODE 2022: 2nd International Seminar on Machine Learning, Optimization, and Data Science*, 248–252. <https://doi.org/10.1109/ISMODE56940.2022.10180952>
- García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (1a ed., Vol. 72). Springer International Publishing. <https://doi.org/10.1007/978-3-319-10247-4>
- González Chamorro, R. M., Díaz Padrón, H., Argote Ravelo, L., Santana López, S., Álvarez Porben, S., Torres Antela, M., Cabrera Guerra, C., Villalba Rodríguez, L., Barroso Sosa,

- N. S., & Jaime Novas, A. (2024). Evaluación de la exposición ocupacional a ozono en áreas de ozonoterapia. *Revista Cubana de Salud y Trabajo*, 25(1).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Goshua, A., Akdis, C. A., & Nadeau, K. C. (2022). World Health Organization global air quality guideline recommendations: Executive summary. *Allergy*, 77(7), 1955–1960. <https://doi.org/10.1111/ALL.15224>
- Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., Kim, R., Raman, R., Nelson, P. C., Mega, J. L., & Webster, D. R. (2016). Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA*, 316(22), 2402–2410. <https://doi.org/10.1001/JAMA.2016.17216>
- Hu, W., & Yang, J. (2024). Effect of ambient ozone pollution on disease burden globally: A systematic analysis for the global burden of disease study 2019. *Science of The Total Environment*, 926, 171739. <https://doi.org/10.1016/J.SCITOTENV.2024.171739>
- Hu, Z., Bai, Z., Yang, Y., Zheng, Z., Bian, K., & Song, L. (2019). UAV Aided Aerial-Ground IoT for Air Quality Sensing in Smart City: Architecture, Technologies, and Implementation. *IEEE Network*, 33(2), 14–22. <https://doi.org/10.1109/MNET.2019.1800214>
- Hua, V., Nguyen, T., Dao, M. S., Nguyen, H. D., & Nguyen, B. T. (2024). The impact of data imputation on air quality prediction problem. *PLOS ONE*, 19(9), e0306303. <https://doi.org/10.1371/JOURNAL.PONE.0306303>
- Huang, J., Li, G., Xu, G., Qian, X., Zhao, Y., Pan, X., Huang, J., Cen, Z., Liu, Q., He, T., & Guo, X. (2018). The burden of ozone pollution on years of life lost from chronic obstructive pulmonary disease in a city of Yangtze River Delta, China. *Environmental Pollution*, 242, 1266–1273. <https://doi.org/10.1016/J.ENVPOL.2018.08.021>

- Humphrey, G. B., Maier, H. R., Wu, W., Mount, N. J., Dandy, G. C., Abrahart, R. J., & Dawson, C. W. (2017). Improved validation framework and R-package for artificial neural network models. *Environmental Modelling & Software*, 92, 82–106.
<https://doi.org/10.1016/J.ENVSOFT.2017.01.023>
- IEEE Standards Association. (2023, mayo 16). *The Evolution of Wi-Fi Technology and Standards*. IEEE Standards Association – Beyond Standards.
<https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/>
- INEC. (2010). *Base de Datos-Censo de Población y Vivienda 2010*.
<https://www.ecuadorencifras.gob.ec/base-de-datos-censo-de-poblacion-y-vivienda-2010/>
- Kingma, D. P., & Ba, J. L. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. <https://arxiv.org/pdf/1412.6980>
- Koziel, S., Pietrenko-Dabrowska, A., Wojcikowski, M., & Pankiewicz, B. (2024). High-performance machine-learning-based calibration of low-cost nitrogen dioxide sensor using environmental parameter differentials and global data scaling. *Scientific Reports*, 14(1), 1–18. <https://doi.org/10.1038/S41598-024-77214-Y>;SUBJMETA=166,172,4081,639,704,987;KWRD=ELECTRICAL+AND+ELECTRONIC+ENGINEERING,ENGINEERING,ENVIRONMENTAL+IMPACT
- Kumar, S., & Jasuja, A. (2017). Air quality monitoring system based on IoT using Raspberry Pi. *Proceeding - IEEE International Conference on Computing, Communication and Automation, ICCCA 2017, 2017-January*, 1341–1346.
<https://doi.org/10.1109/CCAA.2017.8230005>

- Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
<https://doi.org/10.1038/NATURE14539;SUBJMETA=117,639,705;KWRD=COMPUTER+SCIENCE,MATHEMATICS+AND+COMPUTING>
- Liu, C., Zhang, Y., & Zhou, H. (2021). A Comprehensive Study of Bluetooth Low Energy. *Journal of Physics: Conference Series*, 2093(1), 012021. <https://doi.org/10.1088/1742-6596/2093/1/012021>
- Loachamín Tipán, D. C. (2017). *Diagnóstico de la calidad del aire de la ciudad de Ibarra del periodo 2012-2015* [UCE]. <https://www.dspace.uce.edu.ec/handle/25000/10530>
- Ludermir, T. B., & Yamazaki, A. (2003). Neural Networks for Odor Recognition in Artificial Noses. *Proceedings of the International Joint Conference on Neural Networks*, 1, 143–148. <https://doi.org/10.1109/IJCNN.2003.1223317>
- Lyons, S., Nolan, A., Carthy, P., Griffin, M., & O’Connell, B. (2024). Long-term exposure to PM2.5 air pollution and mental health: a retrospective cohort study in Ireland. *Environmental Health*, 23(1), 54. <https://doi.org/10.1186/s12940-024-01093-z>
- Maier, H. R., Jain, A., Dandy, G. C., & Sudheer, K. P. (2010). Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environmental Modelling & Software*, 25(8), 891–909. <https://doi.org/10.1016/J.ENVSOFT.2010.02.003>
- Martínez, F., Charte, F., Frías, M. P., & Martínez-Rodríguez, A. M. (2022). Strategies for time series forecasting with generalized regression neural networks. *Neurocomputing*, 491, 509–521. <https://doi.org/10.1016/J.NEUCOM.2021.12.028>
- Mecharbat, L. A., Benmeziane, H., Ouarnoughi, H., & Niar, S. (2023). HyT-NAS: Hybrid Transformers Neural Architecture Search for Edge Devices. *Proceedings - 2023 IEEE/ACM International Workshop on Compilers, Deployment, and Tooling for Edge AI, CODAI 2023*, 41–45. <https://doi.org/10.1145/3615338.3618131>

- Mienye, I. D., Swart, T. G., & Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information* 2024, Vol. 15, Page 517, 15(9), 517. <https://doi.org/10.3390/INFO15090517>
- Ministerio de Producción, C. E. I. y P. (2023). *PLAN NACIONAL DE LA CALIDAD 2023*. <https://www.produccion.gob.ec/wp-content/uploads/downloads/2023/05/Plan-Nacional-de-la-Calidad-2023-CIMC-vf.pdf>
- Ministerio del Ambiente. (2015). *Acuerdo N° 97/A - Norma de calidad del aire ambiente o nivel de inmisión (Anexo 4, Libro VI de la Calidad Ambiental, del Texto Unificado de la Legislación Secundaria del Ministerio del Ambiente)*. Registro Oficial Edición Especial N° 387, 4 de noviembre de 2015. <https://www.fao.org/faolex/results/details/es/c/LEX-FAOC155133/>
- Ministerio del Trabajo del Ecuador. (2021). *Remuneración mensual por puesto*. <https://www.telecomunicaciones.gob.ec/wp-content/uploads/downloads/2016/04/literal-c-Remuneracion-mensual-por-puesto.-MARZO.pdf>
- Montesinos López, O. A., Montesinos López, A., & Crossa, J. (2022). Fundamentals of Artificial Neural Networks and Deep Learning. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, 379–425. https://doi.org/10.1007/978-3-030-89010-0_10
- Murad, J. A., Roa Martínez, M. Á., & Roa Martínez, M. R. (2023). Incidencia de la altura para la medición de la calidad del aire de PM10 y PM2,5 como parte del cumplimiento del objetivo y características de los SVCA. *Ingeniería y competitividad*, 25(3). <https://doi.org/10.25100/IYC.V25I3.12828>
- Netlify. (2025). *Netlify official website*. <https://www.netlify.com/>
- Nikhade, S. G. (2015). Wireless sensor network system using Raspberry Pi and zigbee for environmental monitoring applications. *2015 International Conference on Smart*

Technologies and Management for Computing, Communication, Controls, Energy and Materials, ICSTM 2015 - Proceedings, 376–381.

<https://doi.org/10.1109/ICSTM.2015.7225445>

Ojeda-Castillo, V., Murillo-Tovar, M. A., Hernández-Mena, L., Saldarriaga-Noreña, H., Vargas-Amado, M. E., Herrera-López, E. J., & Díaz, J. (2024). Tropospheric NO₂: Anthropogenic Influence, Global Trends, Satellite Data, and Machine Learning Application. *Remote Sensing 2025, Vol. 17, Page 49, 17(1)*, 49.

<https://doi.org/10.3390/RS17010049>

Orellano, P., Reynoso, J., & Quaranta, N. (2021). Short-term exposure to sulphur dioxide (SO₂) and all-cause and respiratory mortality: A systematic review and meta-analysis. *Environment International, 150*, 106434.

<https://doi.org/10.1016/J.ENVINT.2021.106434>

Organización de las Naciones Unidas. (2023). *Objetivos de Desarrollo Sostenible - Ciudades*. <https://www.un.org/sustainabledevelopment/es/cities/>

Organización Mundial de la Salud. (2021). *Directrices mundiales de la OMS sobre la calidad del aire Resumen ejecutivo*. Organización Mundial de la Salud.

<https://iris.who.int/bitstream/handle/10665/346062/9789240035461-spa.pdf>

Perera, C., Barhamgi, M., Meedeniya, D. A., Bonilla, V., Campoverde, B., & Yoo, S. G. (2023). A Systematic Literature Review of LoRaWAN: Sensors and Applications. *Sensors 2023, Vol. 23, Page 8440, 23(20)*, 8440. <https://doi.org/10.3390/S23208440>

Perez, P. (2012). Combined model for PM₁₀ forecasting in a large city. *Atmospheric Environment, 60*, 271–276. <https://doi.org/10.1016/J.ATMOSENV.2012.06.024>

Quijano Parra, A., Quijano Vargas, M. J., & Henao Martínez, J. A. (2010). Caracterización fisicoquímica del material particulado-fracción respirable PM_{2.5} en Pamplona-Norte de

- Santander-Colombia. *Bistua: Revista de la Facultad de Ciencias*, 8, 1–20.
<http://www.redalyc.org/articulo.oa?id=90315226007>
- Restrepo Muñoz, V. P. (2009). *Aplicación y comparación de la metodología de diseño Top Down y Bottom Up* [Universidad EAFIT]. <http://hdl.handle.net/10784/8830>
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E., & Valaee, S. (2017). *Recent Advances in Recurrent Neural Networks*. <https://arxiv.org/pdf/1801.01078>
- Salonen, H., Salthammer, T., & Morawska, L. (2019). Human exposure to NO₂ in school and office indoor environments. *Environment International*, 130, 104887.
<https://doi.org/10.1016/J.ENVINT.2019.05.081>
- Santurtún, A., Rasilla, D. F., Riancho, L., & Zarrabeitia, M. T. (2017). Análisis de la relación entre la enfermedad pulmonar obstructiva crónica y los contaminantes atmosféricos atendiendo al origen y trayectoria de las masas de aire en el Norte de España. *Archivos de Bronconeumología*, 53(11), 616–621.
<https://doi.org/10.1016/J.ARBRES.2017.03.017>
- Seed Studio. (2025). *Seed Studio official website*. <https://www.seedstudio.com/>
- Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 404, 132306.
<https://doi.org/10.1016/J.PHYSD.2019.132306>
- Tascon, M., & Coullaut, A. (2016). Big Data y el Internet de las cosas. Qué hay detrás y cómo nos va a cambiar. *Catarata*, 1.
- Ubilla, C., & Yohannessen, K. (2017). CONTAMINACIÓN ATMOSFÉRICA EFECTOS EN LA SALUD RESPIRATORIA EN EL NIÑO. *Revista Médica Clínica Las Condes*, 28(1), 111–118. <https://doi.org/10.1016/J.RMCLC.2016.12.003>
- Unión Internacional de Telecomunicaciones. (2012). *Descripción general de Internet de los objetos (Recomendación UIT-T Y.2060)*.

- United States Environmental Protection Agency. (2024). *Health Effects of Ozone Pollution*.
<https://www.epa.gov/ground-level-ozone-pollution/health-effects-ozone-pollution>
- Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention Is All You Need*. 1.
<https://arxiv.org/pdf/1706.03762>
- Vazquez Santiago, J., Hata, H., Martinez-Noriega, E. J., & Inoue, K. (2024). Ozone trends and their sensitivity in global megacities under the warming climate. *Nature Communications* , 15(1), 1–15. <https://doi.org/10.1038/S41467-024-54490-W>;TECHMETA=134,141;SUBJMETA=169,172,704,824,895,896;KWRD=ATMOSPHERIC+CHEMISTRY,ENVIRONMENTAL+MONITORING,POLLUTION+REMEDIATION
- Veiga, T., Ljunggren, E., Bach, K., & Akselsen, S. (2021). Blind calibration of air quality wireless sensor networks using deep neural networks. *2021 IEEE International Conference on Omni-Layer Intelligent Systems, COINS 2021*.
<https://doi.org/10.1109/COINS51742.2021.9524276>
- Velázquez de Castro González, Federico. (2012). *La contaminación en España*. Club Universitario,.
http://encore.fama.us.es/iii/encore/record/C__Rb2506900__Scambio%20clim%C3%A1tico%20espa%C3%B1a__P0%2C3__Orightresult__X2__T?lang=spi&suite=cobalt
- Wang, M., Li, H., Huang, S., Qian, Y., Steenland, K., Xie, Y., Papatheodorou, S., & Shi, L. (2021). Short-term Exposure to Nitrogen Dioxide and Mortality: a systematic review and meta-analysis. *Environmental research*, 202, 111766.
<https://doi.org/10.1016/J.ENVRES.2021.111766>

- Wei, Y., Jang-Jaccard, J., Xu, W., Sabrina, F., Camtepe, S., & Boulic, M. (2022). LSTM-Autoencoder based Anomaly Detection for Indoor Air Quality Time Series Data. *IEEE Sensors Journal*, 23(4), 3787–3800. <https://doi.org/10.1109/JSEN.2022.3230361>
- Wilamowski, B. M., & Irwin, J. D. (2018). *Intelligent Systems*. CRC Press.
- World Health Organization. (2021, septiembre 22). *What are the WHO Air quality guidelines?* <https://www.who.int/news-room/feature-stories/detail/what-are-the-who-air-quality-guidelines>
- World Health Organization. (2022). *Calidad del aire ambiente (exterior) y salud*. [https://www.who.int/es/news-room/fact-sheets/detail/ambient-\(outdoor\)-air-quality-and-health](https://www.who.int/es/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health)
- Wu, W., Dandy, G. C., & Maier, H. R. (2014). Protocol for developing ANN models and its application to the assessment of the quality of the ANN model development process in drinking water quality modelling. *Environmental Modelling & Software*, 54, 108–127. <https://doi.org/10.1016/J.ENVSOF.2013.12.016>
- Yadav, V., Yadav, A. K., Singh, V., & Singh, T. (2024). Artificial neural network an innovative approach in air pollutant prediction for environmental applications: A review. *Results in Engineering*, 22, 102305. <https://doi.org/10.1016/J.RINENG.2024.102305>
- Yousaf, S., Shafi, I., & Ahmad, J. (2017). Features Selection and Optimized Neural Network Architecture for Modelling Flows in Solar Collectors. *Proceedings - 2017 International Conference on Frontiers of Information Technology, FIT 2017, 2017-January*, 247–252. <https://doi.org/10.1109/FIT.2017.00051>
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *Communications of the ACM*, 64(3), 107–115. <https://doi.org/10.1145/3446776>

Zhao, F., Zhang, C., & Geng, B. (2024). Deep Multimodal Data Fusion. *ACM Computing Surveys*, 56(9). <https://doi.org/10.1145/3649447>

Zhao, R., Yang, Z., Zheng, H., Wu, Y., Liu, F., Wu, Z., Li, L., Chen, F., Song, S., Zhu, J., Zhang, W., Huang, H., Xu, M., Sheng, K., Yin, Q., Pei, J., Li, G., Zhang, Y., Zhao, M., & Shi, L. (2022). A framework for the general design and computation of hybrid neural networks. *Nature Communications*, 13(1), 3427. <https://doi.org/10.1038/S41467-022-30964-7>

ANEXOS

ANEXO A. Ficha de Requerimiento

Proyecto de titulación: “Sistema IoT de monitoreo de Calidad del aire de concentraciones de NO₂ y O₃ con Redes Neuronales en la ciudad de Ibarra”

Objetivo del análisis: Definir de manera precisa los requisitos del proyecto para garantizar el cumplimiento de los objetivos establecidos.

Lista de Stakeholders

Abreviatura	Significado
StRS	Requerimientos de Stakeholders
SyRS	Requerimientos del Sistema
SrSH	Requerimientos de Arquitectura

Requerimientos de Stakeholders			
Nomenclatura	Requerimiento	Descripción	Prioridad

StRS1	El sistema debe almacenar los datos de NO ₂ y O ₃ .	Los altos niveles de dióxido de nitrógeno (NO ₂) y ozono (O ₃), representan un riesgo significativo para la salud pública. En particular, afectan el sistema respiratorio al dañar el tejido pulmonar, lo que puede derivar en enfermedades graves y aumentar el riesgo de muertes prematuras (Querol, 2012).	Alta
StRS2	El sistema debe proporcionar los niveles de concentraciones de contaminantes en el aire.	Determinar la concentración de los contaminantes permite evaluar su nivel de nocividad para la salud y su impacto en la atmósfera. Algunos estudios indican que una reducción significativa en las emisiones podría contribuir a disminuir la mortalidad (Faccioli, 2018).	Alta
StRS3	El sistema debe almacenar coordenadas GPS de las mediciones realizadas.	El almacenamiento de coordenadas GPS permite realizar análisis espaciales con alta precisión y desarrollar mapas detallados (ESRI, 2023). De este modo, es posible registrar y monitorear la ubicación de los análisis generados por el sistema, garantizando un seguimiento preciso de la información geoespacial.	Media
StRS4	El sistema debe almacenar datos de temperatura y humedad.	Las condiciones meteorológicas influyen en los procesos físicos y químicos que intervienen en la formación de ozono, un contaminante secundario derivado del NO ₂ (Faccioli, 2018).	Media

StRS5	El sistema debe tener conexión inalámbrica para la transmisión de datos.	La selección de protocolos de comunicación para un despliegue de un ecosistema IoT en particular va a depender de los requerimientos y constricciones de su caso de uso (Villa Crespo & Morales Alonso, 2017).	Alta
StRS6	El sistema debe tener alimentación por medio de una batería.	La fuente de alimentación del dispositivo es un factor clave para garantizar su movilidad, por lo que debe contar con una capacidad óptima que asegure su correcto funcionamiento y rendimiento.	Media
StRS7	La interfaz web debe ser intuitiva y fácil de usar.	La interfaz web debe ser intuitiva y de fácil navegación, permitiendo el acceder, visualizar y evaluar los análisis de manera eficiente y precisa.	Alta
StRS8	La interfaz web debe mostrar el estado de concentraciones de NO2 y O3.	Los niveles de concentración de NO2 y O3 deben mostrarse de manera clara y organizada en la interfaz web, permitiendo una visualización comprensible.	Alta
StRS9	Portabilidad para usarse en distintas ubicaciones	El dispositivo debe ser portátil para permitir la realización de análisis en diversas ubicaciones y facilitar la evaluación de los niveles de contaminación en distintos entornos.	Media

Requerimientos del Sistema

Nomenclatura	Requerimiento	Descripción	Prioridad
SyRS1	El sistema debe ser capaz de procesar y transmitir datos en tiempo real.	La comunicación en un sistema IoT se basa en la transmisión y recepción estructurada de datos a través de servicios locales o remotos, utilizando redes diversas pero interoperables para garantizar su integración eficiente (Villa Crespo & Morales Alonso, 2017).	Alta
SyRS2	El sistema debe tener una detección adecuada de los NO2 y O3 para su análisis.	Determinar la concentración de los contaminantes permite evaluar su nivel de nocividad para la salud y su impacto en la atmósfera. Algunos estudios indican que una reducción significativa en las emisiones podría contribuir a disminuir la mortalidad (Faccioli, 2018).	Alta
SyRS3	El sistema debe implementar una red neuronal para detección y predicción de niveles de contaminación del aire	Las redes neuronales multicapa optimizan tareas de clasificación y predicción. Como el Aprendizaje Automático, el Aprendizaje Profundo tiene diversas aplicaciones, destacando la detección de anomalías en ciberseguridad y el modelado de escenarios en entornos IoT e IT (Villa Crespo & Morales Alonso, 2017).	Media

SyRS4	La placa de procesamiento debe incluir pines de comunicación para el intercambio de datos.	El protocolo de comunicación como UART en serie que posibilita la interacción entre distintos componentes de un sistema sin la necesidad de sincronización externa. Su aplicación es especialmente relevante en el ámbito de la seguridad, los sistemas embebidos y IoT, donde se requiere una transmisión de datos eficiente y confiable (Nugraha et al., 2024).	Alta
SyRS5	La placa de procesamiento debe contar con pines I2C.	Para interactuar con dispositivos I2C, la placa de procesamiento debe contar con pines específicos para este protocolo. I2C es un estándar ampliamente utilizado para la comunicación entre dispositivos. Emplea dos líneas, SDA (datos) y SCL (reloj), para facilitar la transferencia bidireccional de información entre ellos. (Monk, 2018).	Media
SyRS6	Alimentación eléctrica mediante el puerto USB o conexión de 3.3-5V	Para garantizar el correcto funcionamiento del sistema, la alimentación debe mantenerse dentro del rango de 3.3V a 5V o suministrarse a través del puerto USB. Cumplir con estos valores es esencial para evitar daños en el regulador de voltaje de la placa, asegurando así la estabilidad y protección del circuito (Schmidt, 2022).	Media

SyRS7	La interfaz web debe ser intuitiva para la interacción del usuario.	La interfaz web debe ser intuitiva y de fácil navegación, permitiendo el acceder, visualizar y evaluar los análisis de manera eficiente y precisa.	Alta
SyRS8	La interfaz web debe mostrar un indicador para el estado de la detección de los contaminantes.	Los niveles de concentración de NO2 y O3 deben mostrarse de manera clara y organizada en la interfaz web, permitiendo una visualización comprensible.	Baja
SyRS9	Facilidad de manejo y entendimiento.	Capacidad de interactuar con un sistema de forma intuitiva, reduciendo la complejidad y mejorando la experiencia del usuario.	Media

Requerimientos de arquitectura			
Nomenclatura	Requerimiento	Descripción	Prioridad
SrSH1	La placa de procesamiento debe ser compatible con los sensores.	La placa de procesamiento debe contar con la compatibilidad necesaria para integrar y gestionar los sensores, asegurando una comunicación eficiente y una correcta interpretación de los datos recopilados (Corona et al., 2016).	Alta
SrSH2	Compatibilidad con tecnologías de conexión a internet.	La compatibilidad con tecnologías de conexión a Internet garantiza que un sistema IoT pueda transmitir y recibir información de manera estructurada, permitiendo la interoperabilidad entre diversas redes de comunicación y	Media

		asegurando la integración con servicios locales y remotos (Villa Crespo & Morales Alonso, 2017).	
SrSH3	Disponibilidad de librerías para la funcionalidad de los sensores y placa de procesamiento.	La disponibilidad de librerías de software especializadas es clave para facilitar el desarrollo de sistemas de sensores y microcontroladores, reduciendo la complejidad del proceso de implementación (Garcia Saura, 2014).	Media
SrSH4	El lenguaje de programación debe ser compatible con los sensores.	El software debe estar desarrollado en un lenguaje de programación compatible con los sensores, garantizando una comunicación eficiente, la correcta adquisición de datos y una integración óptima dentro del sistema.	Alta
SrSH5	El sistema debe contar con un diseño compacto que garantice su portabilidad.	El diseño debe ser portátil y adaptable, permitiendo su instalación en ubicaciones fijas, así como su traslado a distintos entornos para análisis del sistema.	Media
SrSH6	El diseño debe permitir a los sensores tener una adecuada recepción de las variables ambientales.	El diseño debe garantizar que los sensores recopilen variables ambientales de manera precisa y eficiente, optimizando su desempeño en la adquisición de datos.	Alta

SrSH7	La placa de procesamiento debe proporcionar suficiente capacidad de procesamiento.	La placa de procesamiento debe garantizar alto rendimiento en operaciones de funcionamiento y paralelización para la red neuronal, evitando cuellos de botella durante la realización de predicciones (Brown et al., 2020).	Alta
SrSH8	La placa de procesamiento debe contar con tecnología de comunicación inalámbrica.	Entre los protocolos de comunicación inalámbrica utilizados en placas embebidas, WiFi y Bluetooth destacan por su prevalencia en aplicaciones de corto alcance, aunque presentan diferencias significativas en su consumo energético (Villa Crespo & Morales Alonso, 2017).	Alta
SrSH9	Los sensores deben ser compatibles con la placa de procesamiento	Algunos sensores comerciales presentan incompatibilidad con placas embebidas debido a diferencias en los protocolos de comunicación implementados o a la limitación física de puertos de entrada/salida, lo que restringe la escalabilidad de sistemas multi-sensor	Media
SrSH10	Los sensores deben trabajar con voltajes de hasta 5v para eficiencia en el consumo energético.	Aunque la mayoría de los sensores operan en rangos de 3.3 V a 5 V, ciertos diseños, debido a sus requisitos técnicos, pueden demandar fuentes de alimentación externas para asegurar su operatividad óptima (Gazis et al., 2021).	Media

SrSH11	Los sensores deben operar en exteriores.	Algunos sensores ambientales no están diseñados para operar en exteriores, ya que su desarrollo está principalmente orientado al uso en interiores (Andrade, 2024).	Baja
SrSH12	Tarjeta de memoria microSD mínimo 64 Gb.	La tarjeta de memoria debe proporcionar a la placa de procesamiento una capacidad de almacenamiento suficiente para garantizar la instalación y ejecución de los paquetes y librerías necesarias.	Media
SrSH13	Fuente de alimentación de 5v.	Para garantizar el correcto funcionamiento del sistema, la alimentación debe mantenerse dentro del rango de 3.3V a 5V o suministrarse a través del puerto USB. Cumplir con estos valores es esencial para evitar daños en el regulador de voltaje de la placa, asegurando así la estabilidad y protección del circuito (Schmidt, 2022).	media
SrSH14	El IDE o lenguaje de programación deben ser de código abierto.	El IDE o lenguaje de programación debe ser de código abierto y cumplir con los principios del software libre, permitiendo la modificación y personalización del código fuente. Además, debe contar con una comunidad activa de desarrolladores y ofrecer compatibilidad con múltiples plataformas (Jacome Leal, 2023).	Alta

SrSH15	Software fácil de usar y que no requiera una curva de aprendizaje extensa	Un software fácil de usar y que no requiera una curva de aprendizaje extensa debe estar diseñado con un enfoque en la usabilidad y la accesibilidad, minimizando la complejidad y facilitando su adopción de herramientas (Zhou, 2024).	Baja
SrSH16	Compatibilidad de librerías para el uso algoritmos de inteligencia artificial.	Diversas bibliotecas ofrecen compatibilidad para la implementación de algoritmos de inteligencia artificial en distintos lenguajes de programación, facilitando su desarrollo y optimización (Rachmawati et al., 2024).	Media

BIBLIOGRAFIA

Querol, X. (2012). Bases científico-técnicas para un plan nacional de mejora de la calidad del aire: (ed.). Editorial CSIC Consejo Superior de Investigaciones Científicas. <https://elibro.net/es/lc/utnorte/titulos/41672>

Faccioli, C. (2018). La protección de la calidad del aire: especial referencia a la actuación local: (ed.). Publicacions Universitat Rovira i Virgili. <https://elibro.net/es/lc/utnorte/titulos/54925>

Villa Crespo, E. & Morales Alonso, I. (2017). *Ciberseguridad IoT y su aplicación en Ciudades inteligentes*: (1 ed.). RA-MA Editorial. <https://elibro.net/es/lc/utnorte/titulos/230294>

Schmidt, D. (2022). Arduino: curso completo: (2 ed.). RA-MA Editorial. <https://elibro.net/es/lc/utnorte/titulos/222675>

Monk, S. (2018). Ejercicios prácticos con Electrónica: proyectos de electrónica con Arduino y Raspberry Pi: (1 ed.). Marcombo.

<https://elibro.net/es/lc/utnorte/titulos/280023>

Nugraha, S. P. A., & Sunuharjo, L. (2024). Komunikasi Arduino I2C, SPI dan UART. Switch: Jurnal Sains dan Teknologi Informasi, 2(4), 80-85.

Corona Ramírez, L. G. Abarca Jiménez, G. S. & Mares Carreño, J. (2016). Sensores y actuadores: aplicaciones con Arduino: (ed.). Grupo Editorial Patria. <https://elibro.net/es/lc/utnorte/titulos/39464>

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33, 1877-1901.

Gazis, A., & Katsiri, E. (2021). Smart home IoT sensors: Principles and applications a review of low-cost and low-power solutions. *International Journal on Engineering Technologies and Informatics*, 2(1), 19-23.

ANDRADE, F. J. D. L. R. (2024). Diseño de una red de dispositivos inalámbricos para monitorear la calidad del aire en interiores.

García Saura, C. (2014). Estrategias cooperativas de detección y localización de olores con robots y narices artificiales (Bachelor's thesis)

Jacome Leal, J. F. (2023). Análisis comparativo de los lenguajes de Programación Ionic y Xamarin orientados a entornos de desarrollo móvil (Bachelor's thesis, Babahoyo: UTB-FAFI. 2023).

Zhou, C. (2024). Challenges and solutions in cross-platform mobile development: a qualitative study of Flutter and React Native.

Rachmawati, O. C. R., Barakbah, A. R., & Karlita, T. (2024). Programming Language Selection for The Development of Deep Learning Library. *JOIV: International Journal on Informatics Visualization*, 8(1), 434-441.