



Universidad Internacional de La Rioja  
Escuela Superior de Ingeniería y Tecnología - ESIT

Máster Universitario en Industria 4.0

# Sistema de seguimiento-control de la contaminación atmosférica aplicado a la salud

Trabajo fin de estudio presentado por:	Monclús Astudillo, Javier
Tipo de trabajo:	Planteamiento de un proyecto de Industria 4.0
Director/a:	González Pérez, Daniel María
Fecha:	08/02/2023

## Resumen

La contaminación atmosférica está asociada a una serie de efectos perjudiciales para la salud, los principales contaminantes son CO<sub>2</sub>, NO<sub>2</sub>, CO, O<sub>3</sub>, PM<sub>10</sub> y PM<sub>2.5</sub> por lo que es necesario desarrollar un sistema de monitorización y control para que las personas puedan estar informadas en todo momento.

En el presente trabajo se plantea una solución novedosa que implica tres desarrollos: la realización de un dispositivo de adquisición de datos mediante sensores capaces de recibir las concentraciones de los gases y partículas perjudiciales para la salud; un sistema de procesamiento, transmisión y tratamiento de datos, capaz de gestionar, almacenar y transmitir los datos de forma segura; y, finalmente, el desarrollo de un sistema de monitorización y control que permite al usuario estar informado en todo momento con avisos personalizados, mediante una aplicación en su teléfono móvil. El resultado final es un sistema de bajo coste para el seguimiento y control de la contaminación atmosférica.

**Palabras clave:** IoT, Contaminación atmosférica, Salud respiratoria, MQTT, Dashboard

## Abstract

Air pollution have been associated with a big amount of adverse health effects, the main pollutants are CO<sub>2</sub>, NO<sub>2</sub>, CO, O<sub>3</sub>, PM<sub>10</sub> and PM<sub>2.5</sub>. Therefore, it is necessary to develop a monitoring and control system so that people can be aware at any time.

Due to the above, this project proposes a new solution divided into three developments. Firstly, a data acquisition device equipped with sensors that receive the concentrations of gasses and particles harmful to health. Secondly, a system of processing, transmission, and data processing, capable of managing, storing, and transmitting data in a safe way. Finally, the development of a monitoring and control system that keeps users always informed and with customized alerts, through an application on your phone. The result is a low-cost system for monitoring and controlling air pollution.

**Keywords:** IoT, Air pollution, Respiratory Health, MQTT, Dashboard

## Índice de contenidos

1.	Introducción .....	1
1.1.	Motivación .....	1
1.2.	Planteamiento del trabajo .....	2
1.3.	Estructura de capítulos .....	3
2.	Contexto y estado del arte .....	5
2.1.	Calidad del aire y salud pública.....	5
2.1.1.	Situación actual de la contaminación atmosférica.....	5
2.1.2.	Gases y partículas que afectan a la salud .....	7
2.1.3.	Efectos de la contaminación atmosférica sobre la salud .....	11
2.2.	Soluciones actuales para el control de la contaminación atmosférica .....	17
2.2.1.	Dispositivos fijos .....	17
2.2.2.	Soluciones actuales en el mercado .....	19
2.2.3.	Proyectos con nuevas soluciones.....	21
2.3.	Tecnologías relacionadas con el proyecto.....	23
2.3.1.	Habilitadores de la Industria 4.0 .....	23
2.3.2.	Hardware .....	26
2.3.3.	Protocolo y comunicación .....	28
2.3.4.	Herramientas para la realización del proyecto .....	30
2.4.	Conclusiones sobre el estado del arte .....	31
3.	Descripción general de la contribución del TFE .....	33
3.1.	Objetivos .....	33
3.2.	Metodología del trabajo .....	33
3.2.1.	Elección y detalles de la metodología utilizada para el proyecto: SCRUM .....	34
3.2.2.	Metodología y fases para el desarrollo del proyecto.....	36



3.3.	Descripción general de las partes o componentes de la propuesta .....	37
4.	Desarrollo específico de la contribución .....	41
4.1.	Diseño y desarrollo del dispositivo de adquisición de datos .....	41
4.1.1.	Elección de microcontrolador y sensores .....	41
4.1.2.	Esquema del dispositivo .....	44
4.1.3.	Funcionamiento del dispositivo .....	45
4.2.	Desarrollo del sistema de procesamiento, transmisión y tratamiento de datos .....	47
4.2.1.	Protocolo de comunicación .....	47
4.2.2.	Broker Cloud .....	50
4.2.3.	EMQX + INFLUXDB + GRAFANA .....	52
4.3.	Diseño y desarrollo del sistema de monitorización y control .....	59
4.3.1.	Aplicación para el usuario .....	59
4.3.2.	Aplicación de notificación por medio del teléfono móvil .....	63
4.3.3.	Análisis de datos históricos .....	63
4.3.4.	Creación de una aplicación Android Studio para la monitorización y control ...	66
4.3.5.	Implementación y solución para interiores .....	73
4.4.	Presupuesto y análisis económico .....	74
5.	Evaluación y análisis de resultados .....	77
5.1.	Clientes y seguridad .....	77
5.2.	Estructura de control y gestión de los datos .....	79
5.3.	Evaluación de los datos y del sistema .....	80
6.	Conclusiones y trabajos futuros .....	86
6.1.	Líneas de trabajo futuras .....	88
	Referencias bibliográficas .....	89
Anexo A.	Código dispositivo de adquisición de datos .....	98

Anexo B. Código dispositivo ventilación en el hogar .....	108
Anexo C. Código Docker, EMQX, InfluxDB, Telegraf y Grafana .....	111
Anexo D. Código Telegraf configuración .....	112
Anexo E. Diagramas de funcionamiento .....	113
Anexo F. Aplicación propia Android Studio, SISCAS.....	116

## Índice de figuras

Figura 1. <i>Medidores de calidad del aire de la OMS</i> .....	18
Figura 2. <i>Mapa estadístico de la concentración de partículas según la OMS</i> .....	18
Figura 3. <i>Mapa estadístico de calidad del aire de la EEA</i> .....	19
Figura 4. <i>Series 500 – Portable Indoor Air Quality Monitor</i> .....	19
Figura 5. <i>Temtop M2000C 2nd y modelos Temtop</i> .....	20
Figura 6. <i>Netatmo NHC-EC (izquierda) y Netatmo NWS01-EC (derecha)</i> .....	21
Figura 7. <i>Esquema de la comunicación para la solución de medición a partir del transporte público</i> .....	21
Figura 8. <i>Dispositivo del proyecto de Suecia para la medición de contaminación</i> .....	22
Figura 9. <i>Gráfica monitorización del NO<sub>2</sub> del dispositivo</i> .....	22
Figura 10. <i>Dispositivo de medición de calidad del aire en interiores LoRaWAN</i> .....	23
Figura 11. <i>Arquitectura de estrella MQTT</i> .....	28
Figura 12. <i>Funcionamiento de la metodología SCRUM</i> .....	34
Figura 13. <i>Marco típico de la metodología SCRUM</i> .....	35
Figura 14. <i>Diagrama resumen de las fases del proyecto</i> .....	37
Figura 15. <i>Arquitectura de funcionamiento del sistema</i> .....	39
Figura 16. <i>Cronograma/Diagrama de Gantt del proyecto</i> .....	40
Figura 17. <i>Esquema ESP-32 NodeMCU Developmentboard Ponout Diagram</i> .....	41
Figura 18. <i>Sensores utilizados para el sistema de monitorización y seguimiento</i> .....	42
Figura 19. <i>Alimentación y batería para sensores y microcontrolador</i> .....	43
Figura 20. <i>Esquema hardware con EasyEDA</i> .....	44
Figura 21. <i>Hardware del dispositivo</i> .....	45
Figura 22. <i>Diagrama inicial resumen del funcionamiento</i> .....	46
Figura 23. <i>Diagrama funcionamiento del Sensor MQ135 en el microcontrolador</i> .....	47

Figura 24. <i>Código para la lectura de valores de topics para el dispositivo</i> .....	49
Figura 25. <i>Arquitectura solución Servidor Cloud MyQttHub</i> .....	50
Figura 26. <i>Características del Servidor gratuito de MyQttHub</i> .....	50
Figura 27. <i>Características del Servidor gratuito de Adafruit</i> .....	51
Figura 28. <i>Broker MyQttHub recibiendo datos del cliente</i> .....	52
Figura 29. <i>Arquitectura del proyecto</i> .....	53
Figura 30. <i>Docker-compose código servicio EMQX</i> .....	54
Figura 31. <i>Docker-compose código servicio InfluxDB</i> .....	54
Figura 32. <i>Docker-compose código servicio Telegraf</i> .....	55
Figura 33. <i>Docker-compose código configuración Telegraf</i> .....	55
Figura 34. <i>InfluxDB datos de temperatura</i> .....	56
Figura 35. <i>Documento csv para la creación de usuarios y contraseñas seguras</i> .....	56
Figura 36. <i>Usuarios y contraseñas del Broker MQTT</i> .....	57
Figura 37. <i>Configurar openssl.cnf para crear solicitud del certificado del servidor MQTT</i> .....	58
Figura 38. <i>Conectar el dispositivo móvil al servidor MQTT</i> .....	59
Figura 39. <i>Panel Dashboard para añadir datos del usuario (Nombre, edad, patologías...)</i> .....	60
Figura 40. <i>Panel Dashboard para monitorización de la contaminación, tiempo y avisos led</i> ..	61
Figura 41. <i>Dashboard datos usuarios y ubicación (izquierda) y configurar panel ubicación (derecha)</i> .....	61
Figura 42. <i>Panel Dashboard temperatura y humedad</i> .....	62
Figura 43. <i>Panel Dashboard monitorización de gases</i> .....	62
Figura 44. <i>Activar notificaciones para un topic específico</i> .....	63
Figura 45. <i>Datos de CO del dispositivo en InfluxDB</i> .....	64
Figura 46. <i>Configurar servicio Grafana (izquierda) y obtener datos de InfluxDB (derecha)</i> ....	64

Figura 47. <i>Configurar panel de gráfico temporal de Grafana a partir de los datos de InfluxDB</i> .....	65
Figura 48. <i>Configurar panel de gauge de Grafana a partir de los datos de InfluxDB</i> .....	65
Figura 49. <i>Dashboard de Grafana</i> .....	66
Figura 50. <i>Gradle para el desarrollo de la aplicación</i> .....	67
Figura 51. <i>AndroidManifest.xml de la aplicación desarrollada</i> .....	67
Figura 52. <i>Código para la conexión MQTT de la aplicación</i> .....	68
Figura 53. <i>Funciones de publicación (izquierda) y suscripción (derecha)</i> .....	69
Figura 54. <i>Layout de la medición de contaminación</i> .....	69
Figura 55. <i>Programación de la actividad principal</i> .....	70
Figura 56. <i>Programación de notificaciones</i> .....	70
Figura 57. <i>Layouts, autenticación y datos de usuarios en Firebase</i> .....	71
Figura 58. <i>Muestra para definir colores (izquierda), definir icono de la aplicación (derecha)</i> .....	72
Figura 59. <i>Capturas de la aplicación, menú e interface de usuario</i> .....	72
Figura 60. <i>Solución para la ventilación en interiores</i> .....	73
Figura 61. <i>Configurar usuario y certificados SSL/TLS para el dispositivo (izquierda) y IoT MQTT Panel (derecha)</i> .....	77
Figura 62. <i>Enviar información MQTT sin SSL/TLS</i> .....	78
Figura 63. <i>Enviar información MQTT con SSL/TLS (encriptado)</i> .....	78
Figura 64. <i>EMQX clientes conectados</i> .....	79
Figura 65. <i>InfluxDB tabla de datos de CO<sub>2</sub></i> .....	79
Figura 66. <i>InfluxDB modos representativos de datos</i> .....	80
Figura 67. <i>Datos de contaminación en una ubicación cercana a donde se han realizado las pruebas</i> .....	80
Figura 68. <i>Comprobación valores de algunos de los sensores</i> .....	81
Figura 69. <i>Comprobación de funcionamiento de notificación y modificación del código</i> .....	82

Figura 70. <i>Código de funcionamiento del cálculo de tiempo</i> .....	82
Figura 71. <i>Comprobación de funcionamiento del cálculo de tiempo</i> .....	83
Figura 72. <i>Aplicación SISCAS evaluación registro datos usuario (izquierda) y notificación (derecha)</i> .....	83
Figura 73. <i>Capturas de la aplicación propia SISCAS</i> .....	84
Figura 74. <i>Prototipo SISCAS, tapa abierta (izquierda), tapa cerrada (derecha)</i> .....	85

## Índice de tablas

Tabla 1. <i>Niveles de AQG recomendados a largo y corto plazo y objetivos</i> .....	9
Tabla 2. <i>Niveles de AQG recomendados a corto plazo</i> .....	10
Tabla 3. <i>Valores límite de los gases contaminantes</i> .....	10
Tabla 4. <i>Fuentes de contaminación atmosférica en los microentornos urbanos</i> .....	16
Tabla 5. <i>Comparación Microcontroladores</i> .....	26
Tabla 6. <i>Sensores actualmente en el mercado para detectar contaminantes atmosféricos</i> ...	27
Tabla 7. <i>Comparación de baterías</i> .....	43
Tabla 8. <i>Comparación de características servidores gratuitos</i> .....	51
Tabla 9. <i>Presupuesto para la construcción del prototipo</i> .....	74
Tabla 10. <i>Previsión del presupuesto 1r año</i> .....	75
Tabla 11. <i>Previsión del presupuesto 2º año</i> .....	76

## Acrónimos y Abreviaturas

AMQP → Advanced Message Queuing Protocol

CO → Monóxido de Carbono

CO<sub>2</sub> → Dióxido de Carbono

CoAP → Constrained Application Protocol

COV → Compuestos Orgánicos Volátiles

EPOC → Enfermedad Pulmonar Obstructiva Crónica

GPIO → Pin de conexión de entrada y salida

IOT → Internet of Things

IP → Protocolo de Internet

M2M → Máquina a máquina

MQTT → Message Queue Telemetry Transport

NH<sub>3</sub> → Amoniac

NO<sub>2</sub> → Dióxido de Nitrógeno

O<sub>3</sub> → Ozono

ppm → Parte por millón

QoS → Calidad de Servicio

SHA → Secure Hash Algorithm (Algoritmo de Hash Seguro)

SISCAS → Sistema de Seguimiento-control de la Contaminación Aplicado a la Salud

SO<sub>2</sub> → Dióxido de Azufre

SSL → Secure Sockets Layers (Capa de sockets seguro)

TCP → Protocolo de control y transmisión

TLS → Transport Layer Security (Seguridad de la capa de transporte)

WHO/OMS → World Health Organization/ Organización Mundial de la Salud

XMPP → Extensible Messaging and Presence Protocol



## 1. Introducción

En este proyecto se desarrolla, mediante los conocimientos adquiridos en el Máster de Industria 4.0, el diseño de un sistema de seguimiento y control de la contaminación atmosférica en los lugares donde el usuario transita, mediante sensores se detecta la concentración de los contaminantes y esta información se transmite al teléfono móvil, utilizando el protocolo de comunicación MQTT (Message Queue Telemetry Transport). Permitiendo al usuario el seguimiento y la recepción de avisos personalizados mediante una aplicación, en la que anteriormente se habrán marcado los parámetros/perfiles para cada usuario. Se trata de una solución de bajo coste, asequible para gran parte de la población.

### 1.1. Motivación

Actualmente, la contaminación atmosférica es un problema de gran importancia, generando efectos dañinos a corto, medio y largo plazo. Las concentraciones actuales de contaminación atmosférica junto a los tiempos de exposición a los mismos se han asociado a una serie de efectos perjudiciales para la salud, generalmente en muertes relacionadas con enfermedades cardiovasculares y respiratorias.

La contaminación atmosférica es una de las principales causas de morbilidad/mortalidad y el motivo principal de este trabajo. Los principales contaminantes que ponen en riesgo la salud son gases y partículas como el Dióxido de Carbono (CO<sub>2</sub>), Ozono (O<sub>3</sub>), Monóxido de Carbono (CO) y Dióxido de Nitrógeno (NO<sub>2</sub>) y partículas que se clasifican en función de su tamaño PM<sub>10</sub> para partículas gruesas (diámetro inferior a 10 µm) y PM<sub>2.5</sub> partículas finas (diámetro inferior a 2,5 µm). (World Health Organization, 2021).

Las enfermedades más comunes relacionadas con la contaminación ambiental son la neumonía, bronquitis crónica, EPOC (enfermedad pulmonar obstructiva crónica), cardiopatía isquémica, asma, asma bronquial, cáncer de pulmón y accidente cerebrovascular, entre otros. (Edwards. et al., 2021).

Por todo lo expuesto es necesario cumplir los parámetros recomendados por la WHO Global Air Quality Guidelines para mantener un buen nivel de salud. (World Health Organization, 2021).

Debido a esta necesidad nace el presente proyecto, donde se desea que cada persona tenga un control y seguimiento de la contaminación atmosférica en los distintos lugares donde se ubica o transita, con información a tiempo real y avisos personalizados.

## 1.2. Planteamiento del trabajo

El aumento constante de la contaminación atmosférica es una preocupación para muchas personas, y por ello es importante saber si realmente el aire que se respira pone en riesgo la salud respiratoria del usuario. (World Health Organization, 2021).

En este trabajo se plantea un sistema capaz de realizar el seguimiento y control de la contaminación atmosférica, donde se miden los principales gases y partículas que ponen en riesgo la salud de las personas.

El sistema está compuesto por un dispositivo de adquisición de datos formado principalmente por sensores para la realización de las mediciones de contaminantes y un microcontrolador que permite enviar y recibir la información. Estos datos se transmitirán mediante el protocolo MQTT, protocolo de código abierto, ligero y flexible, muy adecuado para el uso de sensores, utilizando una conexión TCP/IP (Protocolo de control y transmisión/Protocolo de Internet) que permite tener una comunicación fluida con los clientes publicadores y suscriptores. También incluirá un sistema de procesamiento, transmisión y tratamiento de datos, capaz de gestionar, almacenar y transmitir los datos de forma segura, para el almacenamiento será necesario tener una base de datos, con las mediciones de los sensores y así tener un registro de los datos históricos del dispositivo, se deberán incluir medidas de ciberseguridad como autenticación, autorización y cifrado de datos para la seguridad del sistema. Para finalizar se desarrollará un sistema de monitorización y control que permita al usuario estar informado en todo momento y recibir avisos personalizados, mediante el desarrollo de una aplicación para el teléfono móvil. A fin de dar un servicio global a la sociedad la aplicación permitirá crear el perfil del usuario para así conseguir avisos personalizados, también podrá diferenciar entre las distintas ubicaciones, según se trate de interiores o exteriores, entre otras características. El sistema permitirá detectar en tiempo real si los valores son superiores a los recomendados según la WHO, con su visualización y al mismo tiempo recibir avisos personalizados, para que pueda prevenir o mitigar posibles riesgos por contaminación atmosférica.

### 1.3. Estructura de capítulos

El presente trabajo estará estructurado en diferentes capítulos, a continuación, se detalla el contenido de cada uno de ellos:

#### **1. Introducción**

En este primer capítulo, del que forma parte este apartado de estructura de capítulos, se introduce la temática y problemática asociada a la contaminación atmosférica aplicada a la salud de las personas. Indicando la motivación para la realización de este trabajo junto a una explicación del planteamiento realizado, para abordar el proyecto y así desarrollar la solución.

#### **2. Contexto y estado del arte**

Este segundo capítulo estudia el contexto y estado del arte del proyecto, se presentará el marco teórico en relación con la temática de la que se ha partido para el desarrollo del proyecto. Se indicará la situación actual de la contaminación atmosférica, los gases y partículas contaminantes y dañinos para las personas y sus efectos en la salud.

También se analizarán las soluciones actuales para el control de la contaminación atmosférica, soluciones que existen actualmente en el mercado y proyectos relacionados, con un resumen de sus principales características, destacando sus puntos fuertes y débiles, llegando a identificar mejoras que aportarían un valor añadido para el usuario. También se analizarán e indicarán las tecnologías habilitadoras asociadas al proyecto, detallando las principales tecnologías y herramientas que se utilizarán.

Finalmente, en base a toda la información anterior, se definirán unas conclusiones.

#### **3. Objetivos y metodología de trabajo**

En este capítulo se definen los objetivos del proyecto, tanto el objetivo general como los objetivos específicos. Teniendo identificado lo que se quiere conseguir con la realización del presente proyecto.

También es muy importante definir la metodología que se realizará para el desarrollo de la solución, así se tendrá claro todo el proceso de elaboración para tener un trabajo continuo que aporte valor al proyecto.

#### **4. Desarrollo de la solución**

Este capítulo se centrará en la elaboración detallada de la descripción de cada uno de los elementos que forman la solución. Iniciando con los detalles del diseño y desarrollo del dispositivo de adquisición de datos, el sistema de procesamiento, transmisión, tratamiento y seguridad de los datos y el sistema de monitorización y control.

Finalmente será necesario un presupuesto y análisis del coste la solución, mediante una tabla de presupuesto que indique costes reales de implementación. Junto a un modelo de negocio.

#### **5. Evaluación y análisis de resultados**

En este capítulo se analizarán los resultados obtenidos en la solución por lo que será necesario una fase de evaluación y análisis de la solución, para su demostración se usarán imágenes y capturas para mostrar el correcto funcionamiento del sistema.

#### **6. Conclusiones y líneas de trabajo futuras**

En este capítulo se explicarán las aportaciones obtenidas con los distintos desarrollos de la solución. Dando especial importancia al capítulo anterior de evaluación y análisis de resultados. Finalmente se describirán las líneas de trabajo futuras que pueden llegar a aportar mayor valor añadido a la solución propuesta, junto a posibles soluciones para implementar a nivel comercial este dispositivo.

#### **7. Referencias bibliográficas**

En este último capítulo se indicará un listado de los artículos académicos, revistas científicas, libros y Webs que se han consultado para la realización del proyecto. Estas referencias se han estructurado en formato APA, como se indica en la normativa de la UNIR.

## 2. Contexto y estado del arte

### 2.1. Calidad del aire y salud pública

#### 2.1.1. Situación actual de la contaminación atmosférica

En la Asamblea Mundial de la Salud del año 2015 se concluyó que la contaminación del aire y la salud están directamente relacionadas, y es un factor de riesgo en enfermedades como la cardiopatía isquémica, el ictus, EPOC, el asma y el cáncer. La WHO aporta directrices a los responsables políticos para mejorar la calidad del aire y así mitigar los efectos en la salud y economía. (World Health Organization, 2021).

Las partículas y gases contaminantes que más afectan a la salud son PM<sub>2.5</sub>, PM<sub>10</sub>, Ozono, Dióxido de Nitrógeno, Dióxido de Azufre, Monóxido de Carbono, entre otros. La exposición a contaminantes atmosféricos depende directamente de sus concentraciones en el ambiente. Es importante destacar que en 2019 más del 90% de la población mundial vive en zonas de concentraciones que superaban los valores directrices de la OMS 2015. Según los datos, los patrones de contaminación son diferentes. En el caso de PM<sub>2.5</sub> y Ozono, las mayores concentraciones están en el este de Asia, norte de África, Oriente Medio, Norteamérica y gran parte de Europa, en cambio en el caso del Dióxido de Nitrógeno se observan concentraciones más altas en zonas urbanas en comparación con las zonas rurales. La contaminación atmosférica se origina tanto de fuentes naturales como antropogénicas, siendo las antropogénicas las que más predominan. (World Health Organization, 2021).

En este aspecto se debe diferenciar entre contaminación exterior (outdoor) o contaminación interior (indoor). En la **contaminación en exteriores**, los procesos de combustión son los que más contribuyen a la contaminación del aire en partículas, precisamente el de combustibles fósiles y biomasa para generar energía. Entre las fuentes de combustión se puede destacar: tráfico terrestre, tráfico aéreo, tráfico acuático, industria, generación de combustible, incendios controlados o no controlados, quema de residuos agrícolas o por supuesto residuos de zonas urbanas. También hay que destacar otras fuentes como el polvo superficial, residuos generados por actividades de construcción, entre otros. Otro factor a tener en cuenta es el propio transporte atmosférico, donde el viento o corrientes de aire transportan contaminantes de fuentes lejanas. Con la emisión de contaminantes directos como los PM o a través de procesos fisicoquímicos complejos (para nitratos, sulfatos...), procesos de

agricultura o procesos naturales de emisiones biogénicas. La biomasa en calefacción doméstica es una gran fuente de contaminación exterior. (World Health Organization, 2021).

La **contaminación en interiores**, el uso de combustibles en estufas, calefacción, cocina, velas, incienso, lámparas de queroseno o tabaco, crea compuestos volátiles y semivolátiles como Compuestos Orgánicos Volátiles (COV) y Ozono. También destacar los productos de uso diario como productos de limpieza, que generan amoníaco, insecticidas, aparatos eléctricos o el polvo generado por el movimiento humano. En el interior también se puede observar contaminantes del exterior que entran por la ventilación. Por ese motivo es muy importante la ubicación, por ejemplo, tener en cuenta si está cerca de carreteras, lugares con mucho tráfico, entre otros, se ve un claro aumento de la contaminación que afecta el interior del hogar. Entre los contaminantes de interior se destaca  $PM_{2.5}$ ,  $PM_{10}$ , Amoníaco, Monóxido de Carbono, Dióxido de Nitrógeno, Dióxido de Azufre, Ozono y los contaminantes atmosféricos orgánicos. (World Health Organization, 2021). Otro factor a tener en cuenta en interiores es en relación con los virus, por ejemplo, el SARS-CoV2 se ve influenciado por la contaminación del aire, pudiéndose demostrar que en interiores presenta una alta transmisión. En general para los virus es importante tener en cuenta la exposición de la contaminación interior, en especial de las partículas  $PM_{2.5}$  y el  $NO_2$ , que pueden aumentar la vulnerabilidad y mortalidad a causa de un virus. Por lo que el índice de calidad de aire es una parte fundamental para la prevención de la salud pública y la protección contra la transmisión en una pandemia. (Patil et al.,2022)

Para determinar cómo puede afectar un contaminante a la salud hay dos parámetros que son de especial interés. El primer parámetro se trata de las concentraciones, hace referencia a la proporción o relación del contaminante en relación con un volumen. El segundo parámetro de interés es el tiempo de exposición de los contaminantes, la exposición por un largo periodo para contaminantes de bajas concentraciones también se ha asociado a la mortalidad, en especial las partículas  $PM_{2.5}$  y el  $NO_2$ . (Stafoggia et al.,2022) Estos dos parámetros determinarán los limitantes de exposición a un contaminante. (World Health Organization, 2021).

Según un estudio de la OMS en 2018, alrededor de 7 millones de muertes se deben principalmente por enfermedades no transmisibles que son atribuidas a los efectos de la contaminación ambiental y doméstica. Generando un impacto en la salud y en la economía,

donde poblaciones con menos recursos sufren más muertes por las enfermedades generadas por ello. (OMS,2018)

Evaluaciones globales sugieren que la contaminación genera millones de muertes y cientos de millones de años de vida perdidos, principalmente relacionadas con las PM<sub>2.5</sub>, con enfermedades, como son infecciones agudas de las vías respiratorias, EPOC, cardiopatía isquémica CI, cáncer de pulmón y accidente cerebrovascular (Cohen et al.,2017) (OMS,2018).

### 2.1.2. Gases y partículas que afectan a la salud

Entre los gases contaminantes que más afectan se puede destacar:

- **Dióxido de Carbono (CO<sub>2</sub>):** Principalmente emitido por el tráfico de vehículos y seres vivos. En concentraciones elevadas puede causar dolores de cabeza, falta de concentración, somnolencia, mareos o problemas respiratorios. Para personas con problemas de asma es conveniente que respiren aire con bajas concentraciones de CO<sub>2</sub>. Hasta los 400 ppm (partes por millón) la concentración sería adecuada y empezaría a ser peligroso a partir de 1000 ppm y muy peligroso a partir de 1500 ppm. En interiores recomiendan valores inferiores de 500 hasta 700 ppm. (Ayuntamiento de Barcelona, 2021)

- **Ozono (O<sub>3</sub>):** No emitido directamente, el O<sub>3</sub> se forma en la atmosfera por reacciones entre óxidos de nitrógeno y compuestos orgánicos volátiles en presencia de calor y luz solar. Debido a este motivo es bastante frecuente a España como en otras zonas del sud de Europa (por la elevada insolación de esas regiones, se ve aumentado con el incremento de radiación solar) Es un gas incoloro que se forma en una capa cercana al suelo por la reacción química de ciertos contaminantes. Sin embargo, a cotas inferiores, en la troposfera (la capa de la atmósfera en contacto con la tierra), donde se respira, se convierte en un contaminante que actúa como un potente y agresivo agente oxidante. Entre los efectos dañinos para la salud se puede destacar la generación de problemas respiratorios envejecimiento prematuro de los pulmones, irritación, ataques de asma, tos, inflamación bronquios, bronquitis aguda, enfisema, enfermedades pleurales, insuficiencias respiratorias, asma e incrementa a la mortalidad prematura. Entre los grupos de riesgo se destaca personas con enfermedades respiratorias crónicas, niños, adultos que realizan una activada intensa física exterior, debido al aumento de frecuencia respiratoria y finalmente personas alérgicas o especialmente sensibles a este gas. En este tipo de población no es recomendable la presencia de este gas cuando supera los

180  $\mu\text{m}/\text{m}^3$ . En el resto de población es recomendable no superar los 240  $\mu\text{m}/\text{m}^3$ . Entre los parámetros a tener en cuenta están el tiempo y la concentración. Finalmente, como información adicional en el interior la concentración del Ozono es de aproximadamente el 50% del exterior, debido al sol. (Sanitas, 2021) (MITECO, 2021)

- **Monóxido de Carbono (CO):** Producto de la combustión incompleta del gas natural, carbón o madera (carburantes fósiles y biocombustibles). En las ciudades principalmente generado por el tráfico de vehículos. A nivel de salud penetra en el organismo por los pulmones y provoca una disminución de capacidad de transporte de oxígeno de la sangre, por lo que baja la cantidad de oxígeno destinada a los órganos y tejidos por lo que puede provocar anomalías cardíacas y daños en el sistema nervioso. (MITECO, 2021)

- **Dióxido de Nitrógeno (NO<sub>2</sub>):** Generado por el ser humano, se emite principalmente como resultado de la combustión a alta temperatura, como sería el caso de empresas eléctricas, calderas, industrias y en especial el tráfico. El tráfico de vehículos es un de las principales fuentes de generación en las ciudades, por ese motivo las grandes acumulaciones de este gas contaminante se encuentran en grandes ciudades.

El óxido de nitrógeno se encuentra en dos formas distintas NO<sub>2</sub> y NOX (NO+NO<sub>2</sub>), destacar que la forma principal es el NO<sub>2</sub>, la forma química con efectos dañinos para la salud, además, el NO se oxida con facilidad y rápidamente se convierte en NO<sub>2</sub>. Sus efectos en la salud son inflamación de las vías respiratorias, efectos dañinos a los órganos entre ellos destacar el hígado y bazo y también daña a sistemas como es el caso del sistema circulatorio y sistema inmunitario, este último haciéndolo favorable para la generación de infecciones pulmonares e insuficiencias respiratorias. También se ha observado una mayor incidencia de bronquitis, sobre todo a inmunodeprimidos y bronquiolitis en niños. Hay estudios que lo relacionan con un aumento de las posibilidades de partos prematuros o recién nacidos de peso inferior al promedio (Sram et al., 2013). Son especialmente sensibles los niños menores de 6 años, embarazadas, mayores de 65 años y pacientes oncológicos. También personas con asma, enfermedades respiratorias crónicas o problemas cardiovasculares. (MITECO, 2021)

- **Partículas PM<sub>10</sub>** (partículas gruesas, diámetro inferior a 10  $\mu\text{m}$ ): Proviene de la suspensión de polvo, tierra, polen, moho, esporas y otros materiales. (MITECO, 2021)



- **Partículas PM<sub>2.5</sub>** (partículas finas, diámetro inferior a 2,5 µm): Provenientes de emisiones en procesos de combustión, como en vehículos de gasolina y diésel. También cuando se usa la combustión como fuente de energía y procesos industriales. En otros casos se producen en la atmosfera como resultado de las reacciones químicas de los gases precursores (SO<sub>2</sub>, NO<sub>x</sub>, NH<sub>3</sub> y compuestos orgánicos volátiles). Este tipo de partículas son las que dañan más el cuerpo, principalmente los pulmones y corazón, penetran hasta lo más profundo del sistema respiratorio y afectando al flujo sanguíneo. Las principales afecciones que generan son muerte prematura por enfermedades cardíacas o pulmonares, infartos, asma agudo, reducción de la función pulmonar con irritación de las vías respiratorias. Entre las personas que más afectan serían personas con enfermedades cardíacas o pulmonares, niños o personas mayores de 65 años. (MITECO, 2021) (EPA, 2021). En la Tabla 1 se muestra un resumen de las directrices de calidad del aire según la Organización Mundial de la Salud.

**Tabla 1.** Niveles de AQG recomendados a largo y corto plazo y objetivos

Contaminantes	Promedio de Tiempo	Objetivos intermedios				AQG nivel
		1	2	3	4	
<b>PM<sub>2.5</sub></b> (µg/m <sup>3</sup> )	Anual	35	25	15	10	5
	24 horas	75	50	37,5	25	15
<b>PM<sub>10</sub></b> (µg/m <sup>3</sup> )	Anual	70	50	30	20	15
	24 horas	150	100	75	50	45
<b>O<sub>3</sub></b> (µg/m <sup>3</sup> )	Temporada alta	100	70	-	-	60
	8 horas	160	120	-	-	100
<b>NO<sub>2</sub></b> (µg/m <sup>3</sup> )	Anual	40	30	20	-	10
	24 horas	120	50	-	-	25
<b>CO</b> (mg/m <sup>3</sup> )	24 horas	7	-	-	-	4
<b>SO<sub>2</sub></b> (µg/m <sup>3</sup> )	24 horas	125	50	-	-	40

Fuente: World Health Organization, 2021n.

La Tabla 2 muestra las directrices de calidad del aire para el NO<sub>2</sub>, SO<sub>2</sub> y CO para periodos cortos de tiempo:

**Tabla 2.** Niveles de AQG recomendados a corto plazo

Contaminantes	Promedio de Tiempo	AQG level
<b>NO<sub>2</sub></b> (µg/m <sup>3</sup> )	1 hora	200
<b>SO<sub>2</sub></b> (µg/m <sup>3</sup> )	10 min	500
<b>CO</b> (mg/m <sup>3</sup> )	8 horas	10
	1 hora	35
	15 minutos	100

Fuente: World Health Organization, 2021n.

La Tabla 3 muestra una comparación de los valores límites de los gases contaminantes según el valor legislativo de la Unión Europea en comparación del valor indicado en la guía de la OMS.

**Tabla 3.** Valores límite de los gases contaminantes

Contaminantes	Valor legislado UE	Valor guía OMS (2006)
<b>CO<sub>2</sub></b> (interiores)	1500 ppm	-
<b>NO<sub>2</sub></b> (horario)	200 µg/m <sup>3</sup> (>18 veces/año)	200 µg/m <sup>3</sup>
<b>NO<sub>2</sub></b> (anual)	40 µg/m <sup>3</sup>	40 µg/m <sup>3</sup>
<b>O<sub>3</sub></b> (máximo diario 8h)	120 µg/m <sup>3</sup>	100 µg/m <sup>3</sup>
<b>CO</b> (máximo diario 8h)	10 mg/m <sup>3</sup>	10 mg/m <sup>3</sup>
<b>PM<sub>10</sub></b> (diario)	50 µg/m <sup>3</sup> (>35 veces/año)	50 µg/m <sup>3</sup> (>3 veces/año)
<b>PM<sub>10</sub></b> (anual)	40 µg/m <sup>3</sup>	20 µg/m <sup>3</sup>
<b>PM<sub>2.5</sub></b> (diario)	-	25 µg/m <sup>3</sup> (>3 veces/año)
<b>PM<sub>2.5</sub></b> (anual)	25 µg/m <sup>3</sup>	10 µg/m <sup>3</sup>

Fuente: Ministerio para la Transición Ecológica y el Reto Demográfico, 2021n.

También hay otros gases, de los que se ha descartado la medición, debido a su poca presencia en núcleos urbanos, que por el contrario tienen presencia en procesos industriales y en la extracción de minerales:

- **Dióxido de Azufre (SO<sub>2</sub>):** Proviene de la quema de combustibles de alto contenido de azufre y petróleo. Principalmente se genera en procesos industriales y extracción de metales a partir de minerales. Debido a este motivo no se encuentra en grandes ciudades. Se puede poner el ejemplo de la Republica Checa que, debido a la importante extracción de metales y minería diversa, tienen altos niveles de este contaminante (Sram R., 2020) (MITECO, 2021)

- **Amoniaco:** Presente sobre todo en productos de limpieza en el hogar (MITECO, 2021)

- Otros metales como Antimonio, Cobre, Zinc y Bario (Sb, Cu, Zn y Ba), entre otros. (MITECO, 2021)

### 2.1.3. Efectos de la contaminación atmosférica sobre la salud

Los principales contaminantes que ponen en riesgo la salud son las partículas que se clasifican en función de su tamaño PM<sub>10</sub> para partículas gruesas (diámetro inferior a 10 µm) y PM<sub>2.5</sub> partículas finas (diámetro inferior a 2,5 µm). Y los gases como el CO<sub>2</sub>, O<sub>3</sub>, CO y NO<sub>2</sub>. (World Health Organization, 2021)

La exposición a la contaminación atmosférica puede clasificarse crónica en caso de que sea constante como es el caso de vivir en ciertas ciudades o aguda en el caso que la persona presente picos de contaminación en un periodo de tiempo corto. (Rusu-Zagar et al., 2013)

Entre las enfermedades generadas por la contaminación del aire se destaca la neumonía, bronquitis crónica, enfermedad pulmonar obstructiva crónica, cardiopatía isquémica, asma, asma bronquial, cáncer de pulmón y accidente cerebrovascular, entre otros. (Khomenko et al., 2021)

También se relaciona con la diabetes tipo II y sobre todo en la mortalidad neonatal por el bajo peso y el período de gestación total. También puede aumentar la incidencia en el Alzheimer y otras enfermedades neurológicas. (Zu et al., 2020)

### 2.1.3.1. Enfermedades causadas por la contaminación del aire

A continuación, se detallan más características de cada una de las enfermedades causadas por la contaminación del aire:

- **Neumonía:** Enfermedad a causa de una infección aguda en los pulmones, una inhalación de aire contaminado duplica el riesgo de sufrir neumonía, especialmente en la niñez y en personas mayores. (Medlineplus, 2021)

- **Enfermedad pulmonar obstructiva crónica (EPOC):** Enfermedad pulmonar inflamatoria que causa la obstrucción del flujo de aire de los pulmones. Los síntomas que genera dificultad para respirar, tos y producción de moco. Es causado por la exposición a gases irritantes o partículas PM. Esta enfermedad aumenta el riesgo a desarrollar enfermedades cardíacas o cáncer de pulmón. Dos afecciones que contribuyen a esta enfermedad son enfisema y la bronquitis crónica. (Mayo Clinic, 2021) (Medlineplus, 2021)

- **Bronquitis crónica:** Inflamación del revestimiento de los bronquios, encargados de llevar el aire de los alveolos a los pulmones. La bronquitis crónica es generada por factores como humo del tabaco, polución o patologías genéticas. (Medlineplus, 2021)

- **Enfisema:** Afección en la que los alveolos de los pulmones se destruyen, las principales causas son la exposición al humo de tabaco, gases y partículas irritantes (Medlineplus, 2021)

- **Asma bronquial / Bronquitis asmática:** Proceso por el cual los diferentes alérgenos o desencadenantes de inflamación crónica de los bronquios, esto dificulta el respirar, es habitual sensación de ahogo sibilancias y tos. (Clínica Universidad de Navarra, 2021)

- **Asma:** Afección en las vías respiratorias donde se estrechan al hincharse, lo que provoca mayor mucosidad, dificultad de respiración y falta de aire. (Medlineplus, 2020)

- **Cardiopatía isquémica:** Estrechamiento u obstrucción de las arterias del corazón, si es de modo lento es una angina de pecho si se tapona rápidamente es un infarto. (Clínica Barcelona, 2021)

- **Ictus/Accidente cerebrovascular:** Obstrucción del flujo sanguíneo en una parte del cerebro. Con consecuencias muy incapacitantes. (Mayo Clinic, 2022)

- **Cáncer de pulmón:** Cáncer ocurre cuando una célula o grupo de celular se multiplica (proliferación) exageradamente y sin control de determinadas células del pulmón. Ese espacio

puede generar compresión a estructuras cercanas, afectar a otros órganos y tejidos del cuerpo llegando a afectar sus funciones. (Mayo Clinic, 2021)

En 2015, se estima que 4,2 millones de muertes a nivel mundial fueron debidas a partículas  $PM_{2.5}$  (atribuidas a la contaminación del aire). Esto equivale a alrededor de 1 de cada 3 muertes a nivel mundial. Entre las causas más frecuentes se distingue accidente cerebrovascular, cardiopatía isquémica, cáncer de pulmón y enfermedad pulmonar obstructiva crónica.

En la contaminación del aire en interiores afecta principalmente a personas con enfermedades cardiovasculares y respiratorias (preexistentes). Entre otros factores que aporta una calidad de aire interior con niveles no adecuados son una reducción de la productividad y del rendimiento personal. Si la exposición es prolongada es capaz de dañar a los ocupantes. La contaminación interior está relacionada con problemas de inflamación, infecciones respiratorias, asma y alergias. (Yin et al., 2022).

La contaminación atmosférica es una de las grandes causas de mortalidad, generando una gran cantidad de muertes cerebrovasculares y cardiovasculares. Numerosos estudios han demostrado que las personas que viven en zonas con altos niveles de contaminación de manera permanente presentan mayor riesgo de enfermedades cardiovasculares. Un metaanálisis demostró que por cada  $10 \text{ g/m}^3$  aumento en los niveles atmosféricos  $PM_{2.5}$  el riesgo de mortalidad aumentaba un 6% y el riesgo de muerte cardiovascular aumentó un 11%. También se demostró que con el aumento de  $5 \text{ g/m}^3$  se asoció con un aumento en el riesgo de accidente cerebrovascular. Las partículas  $PM_{2.5}$  están asociadas con la estenosis de la arteria carótida. La exposición aguda, muchas veces generados en centros urbanos, donde a veces se producen picos, a esta contaminación del aire se relaciona con problemas vasculares en específico el accidente cerebrovascular.

En un metaanálisis se relacionó que los días donde se registraron mayores niveles de contaminación eran los que se producían más cantidad de accidentes cerebrovasculares. (García A. et al., 2021)

Entre los contaminantes que se encuentran en los edificios de la ciudad de Xi'an, destacan las  $PM_{2.5}$  y  $CO_2$ , Xi'an, una gran ciudad, capital de provincia de Shaanxi, en el centro de China. Se analiza que durante invierno el nivel de contaminación era superior en comparación del

verano, principalmente debido a la poca ventilación durante los meses de invierno por el frío. Otro factor a tener en cuenta era el tipo de ventilación la ventilación natural o ventilación mecánica. Donde en la concentración de CO<sub>2</sub> la ventilación natural es mejor que la mecánica, debido que con ventilación mecánica los ocupantes no abrían las ventanas. En cambio en concentraciones de partículas PM<sub>2.5</sub>, cuando en el exterior esta elevado este contaminante es mejor la ventilación mecánica. (Yin et al., 2022).

A nivel mental se puede reafirmar que una calidad de aire mala está relacionada con una salud mental negativa, sobre todo en momentos de cierta presión como sería el caso de estrés académico, transición de estudiante universitario al mundo laboral, entre otras situaciones. Lo que podría llevar a la depresión, infelicidad hedónica o poca satisfacción por la vida (Zu et al., 2022).

En general las pequeñas partículas y otros contaminantes del humo de interiores inflaman las vías respiratorias y los pulmones, dificulta la respuesta inmunitaria y reducen la capacidad de oxigenación de la sangre.

También se ha demostrado que el asma puede empeorar con la contaminación del aire y ello es debido a que la contaminación del aire dificulta la respiración. Dos de los contaminantes que más pueden afectar al asma son el ozono y la contaminación particular como neblina, humo y polvo, la presencia de estos dos contaminantes favorece la probabilidad de tener los síntomas del asma.

Es necesario destacar los trabajadores o personas que vivan cerca de intersecciones muy concurridas y con mucho tráfico tienen efectos dañinos para su salud. Un estudio que se realizó a trabajadores en tres intersecciones durante julio de 2005, 2006 y 2007 muestra que es necesario tomar medidas para proteger a los trabajadores, debido al tráfico los valores de Ozono superaron el límite de 120 µg/m<sup>3</sup> al igual que las partículas PM<sub>10</sub> superando las 50 µg/m<sup>3</sup>. El valor de las partículas 2.5 también fueron superiores con 2,3 µg/m<sup>3</sup> y 2,4 µg/m<sup>3</sup> y 1,74 µg/m<sup>3</sup>. Todos estos valores generados por el tráfico de vehículos en esas intersecciones concurridas por lo que en estos casos se recomendaba el uso de equipo de protección adecuado (Rusu et al. 2013).

### 2.1.3.2. Efectos en la población o lugares específicos

Entre las principales personas de riesgo se puede destacar:

#### **-Embarazo y recién nacidos:**

- En un análisis de miles de embarazos durante cuatro años se demostró que el primer mes de gestación es el más sensible para la absorción de contaminación ambiental, principalmente de partículas PM<sub>2.5</sub> (Sram R., 2020)
- También se observó un retraso del crecimiento intrauterino (RCIU) por causas asociadas a PM<sub>10</sub> y c-PAHs (hidrocarburos aromáticos policíclicos cancerígenos) en el primer mes de gestación (Sram et al., 2013)
- La combustión de Carbón en los hogares asociado a mayor incidencia de las vías respiratorias y menor crecimiento del esqueleto en la primera infancia hasta los 3 años. (Sram et al., 2013)
- Estudios han demostrado que la contaminación atmosférica afecta a la salud de los recién nacidos. Donde se puede observar un aumento de la morbilidad respiratoria, esto se pudo evaluar gracias al uso de biomarcadores. (Sram et al., 2013)

#### **-Niños:**

- Asma: Hay una relación directa entre el asma y la contaminación atmosférica, sobre todo en los primeros años de vida. Se han demostrado evidencias científicas de los efectos relevantes de la contaminación atmosférica en la salud respiratoria de los niños, tanto por estar más expuestos como por ser más susceptibles (Ubilla et al., 2017).
- Uno de los principales casos de riesgo en altas concentraciones de CO<sub>2</sub>, los niños debido a su elevado metabolismo producen una gran cantidad de CO<sub>2</sub>, por eso es importante regular la ventilación en la casa y en los colegios. (Ubilla et al., 2017)
- Se relacionó el aumento de las concentraciones de PM<sub>2.5</sub> y HAP con el desarrollo de bronquitis para los niños de edad preescolar. (Sram et al., 2013)
- También se asoció una diferencia genética de los leucocitos de los niños que presentaban asma bronquial debido a vivir en una región contaminada. (Sram et al., 2013) (Ubilla et al., 2017)

**- Personas Mayores:**

- Debido al elevado tiempo que pasan en las casas es necesario que se regule la ventilación de sus hogares, a este punto se le añadirá las posibles enfermedades neurodegenerativas por lo que es difícil tener el control de la ventilación.
- Personas con problemas cardiacos y/o respiratorios, o la morbilidad vinculada a otras enfermedades.

También es importante ser capaz de diferenciar entre los entornos donde estás ubicado la mayoría del tiempo. Entre ellos se puede diferenciar el domicilio, el lugar de trabajo, calles y entornos de ocio como gimnasios, cines y parques.

La Tabla 4 muestra los principales microentornos urbanos, junto a los gases contaminantes a los que se está expuesto en cada uno de ellos:

**Tabla 4.** Fuentes de contaminación atmosférica en los microentornos urbanos

Microentornos	Fuentes	Contaminantes
<b>Domicilio</b>	Cocina, calefacción, vehículos aparcados, pasatiempos, fumar, productos domésticos, animales domésticos, roedores e insectos.	PM, CO, COV, NO <sub>x</sub> , alérgenos
<b>Entorno de Transporte</b>	Emisiones de vehículos e industria, polvo de carretera, contaminación del suelo, fumar	PM, CO, NO <sub>x</sub> , O <sub>3</sub> , VOCs, Aero alérgenos, carcinógenos
<b>Calles</b>	Emisiones de vehículos, polvo de carretera y contaminación del suelo	PM, CO, NO <sub>x</sub> , O <sub>3</sub> , COV, Carcinógenos y plomo
<b>Entornos laborales</b>	Procesos industriales, tabaquismo, contaminación	PM, CO, COV, NO <sub>x</sub>
<b>Entornos de ocio</b>	Cocinar, calentar espacios, contaminación de suelo, humo de tabaco	PM, COV, carcinógenos

Fuente: World Health Organization, 2021n.



### 2.1.3.3. Beneficios que se obtienen gracias a medición de la contaminación del aire

La capacidad para la monitorización de los diferentes contaminantes a la que los usuarios están expuestos aporta una información muy valiosa para evitar este tipo de gases y ver si hay zonas o momentos donde el aire afecta de forma dañina a la salud humana. La monitorización permite analizar el cumplimiento de las recomendaciones de la calidad del aire y preparar medidas para mitigar los efectos perjudiciales de la contaminación.

En este proyecto se obtiene un sistema que aporta beneficios como saber la contaminación en tiempo real, esto se puede aplicar cuando es necesaria la ventilación del hogar, si es adecuado salir a pasear, saber la zona donde es recomendable hacer ejercicio entre otros usos. En relación con el SARS-CoV 2 la medición de CO<sub>2</sub> puede llevar beneficios al usuario, para reducir su transmisión.

## 2.2. Soluciones actuales para el control de la contaminación atmosférica

### 2.2.1. Dispositivos fijos

En la actualidad, las soluciones que aportan organismos públicos en relación con la contaminación atmosférica son conocer los valores y gráficas con los distintos contaminantes en ciudades, países o territorios extensos.

Entre las primeras soluciones se pueden destacar medidores fijos ubicados en puntos estratégicos, en este caso se puede destacar, la OMS que tiene ubicados distintos medidores de calidad del aire para tener datos mundiales de la contaminación, tiene sus bases de datos OMS y OpenAQ donde proporcionan información sobre las concentraciones medias anuales. En la Figura 1 se puede observar una fotografía de un modelo de medidores de calidad del aire de la OMS.

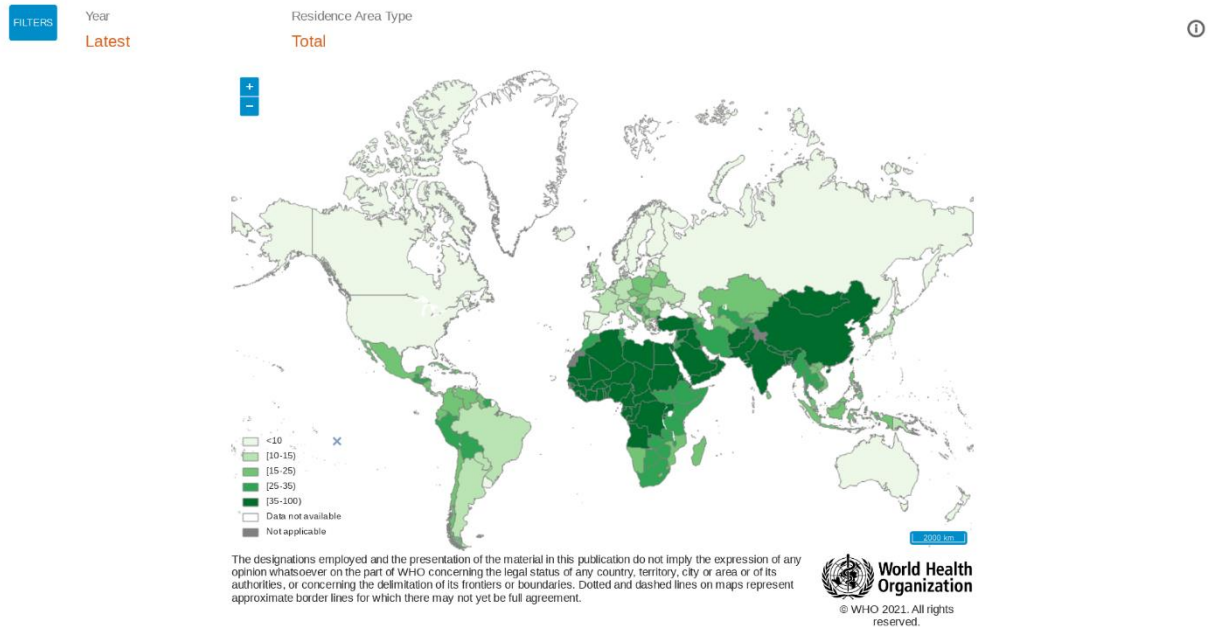


**Figura 1. Medidores de calidad del aire de la OMS**

Fuente: WHO, 2021n.

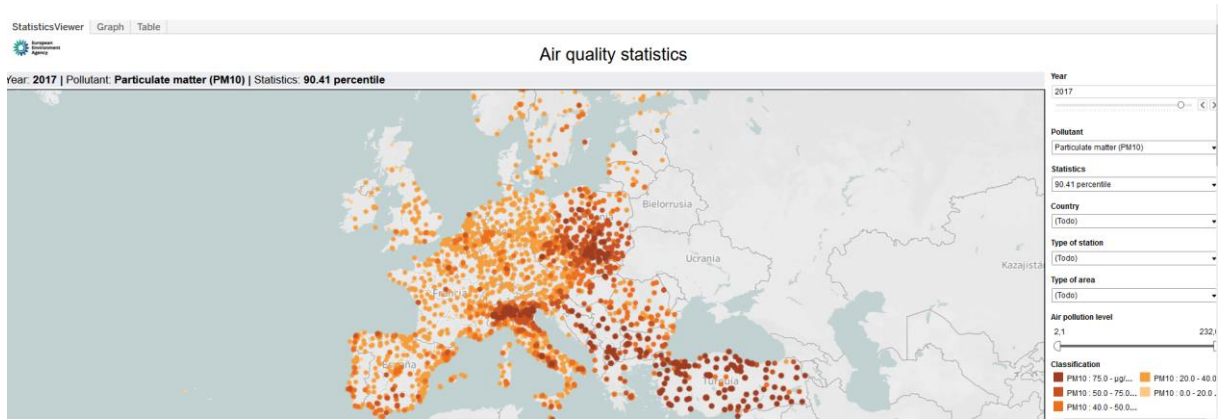
Estos datos se pueden exportar a Excel, aunque WHO también ofrece mapas interactivos donde se muestra valores de contaminación por países, en la Figura 2 se observa un mapa interactivo donde te permite ver la concentración de partículas  $PM_{2.5}$  según cada país. Y en la Figura 3 se muestra la calidad de aire de diferentes países.

Concentrations of fine particulate matter (PM<sub>2.5</sub>)



**Figura 2. Mapa estadístico de la concentración de partículas según la OMS**

Fuente: WHO, 2021o.



**Figura 3.** Mapa estadístico de calidad del aire de la EEA

Fuente: EEA, 2021o.

### 2.2.2. Soluciones actuales en el mercado

Entre las soluciones que se encuentran en el mercado se distingue dos grandes grupos: productos destinados al área industrial u otros más relacionados con el uso privado o particular.

Un ejemplo son los dispositivos de Aeroqual como el medidor de gas AQ-500 este producto es aplicado a nivel industrial y se destaca por su precisión en las mediciones, aunque como principal inconveniente es su elevado precio, alrededor de los 1.403 euros. Entre las características a destacar es su alta precisión, tiene incorporada una pantalla LCD, incluye batería, también una función de alarma sonora y capacidad de conexión a PC con cable para el registro de datos. Entre los gases que es capaz de medir se distingue el O<sub>3</sub>, CO, NH<sub>3</sub>, CO<sub>2</sub>, CH<sub>4</sub> SO<sub>2</sub>, entre otros. Ofrece una gran variedad de medición de gases.

En la Figura 4 se observa una fotografía del modelo:



**Figura 4.** Series 500 – Portable Indoor Air Quality Monitor

Fuente: PCE-instruments, 2021o.

Entre las soluciones más económicas, para consumidores o particulares, existen soluciones por ejemplo el Temtop M2000C se trata de un dispositivo de monitoreo de la calidad del aire, este dispositivo es capaz de monitorear las partículas PM<sub>10</sub>, PM<sub>2.5</sub>, CO<sub>2</sub>, temperatura y humedad. Incluye alarma de audio y batería. Se trata de una solución más económica a precio de aproximadamente unos 200 euros, su principal inconveniente es que no mide otros gases muy importantes para la salud como es el caso de los gases Ozono, muy presente en nuestro país, NO<sub>2</sub> o CO. También hay que tener en cuenta es que el dispositivo no tiene conexión móvil por donde transmitir los datos ni generación de avisos personalizados. En la Figura 5 se observa el dispositivo y las características de las diferentes gamas.

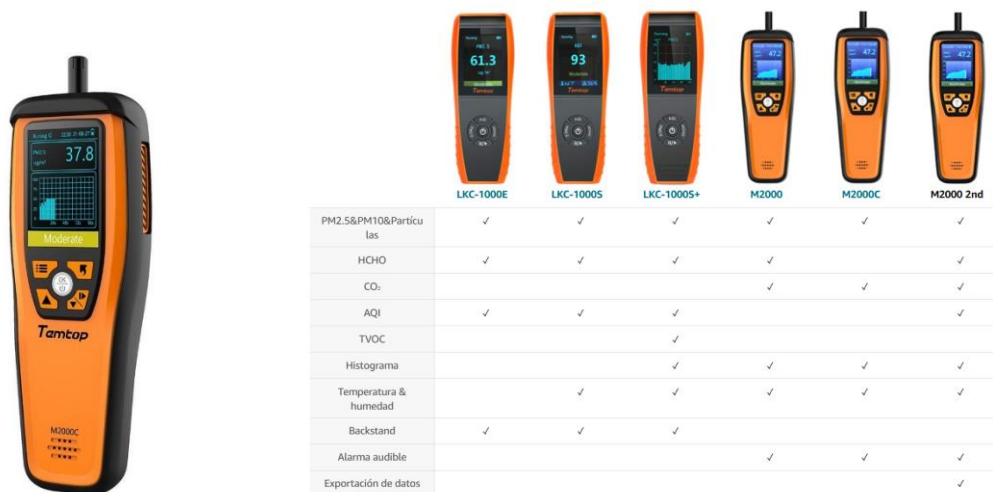


Figura 5. Temtop M2000C 2nd y modelos Temtop

Fuente: Temtop, 2021o.

Otros modelos como el Netatmo NHC-EC capaz de tener el dispositivo conectado con el teléfono móvil, pero sólo ofrece mediciones básicas como serían el CO<sub>2</sub>, temperatura, humedad, por todo ello está más destinado al interior del hogar. Su precio rondaría los 120 euros. Entre otros productos de la marca, destaca el modelo Netatmo NWS01-EC que ofrece más una estación meteorológica, con capacidad de medir CO<sub>2</sub> y tiene un control de la calidad de aire más genérico, donde no se centra en qué tipo de gases pueden dañar, ni genera avisos. Por lo que gases o partículas que son nocivos para el cuerpo no se miden, entre los que no mide este dispositivo se distingue el Ozono o el dióxido de nitrógeno.

En la Figura 6 se observa una imagen con estos modelos:



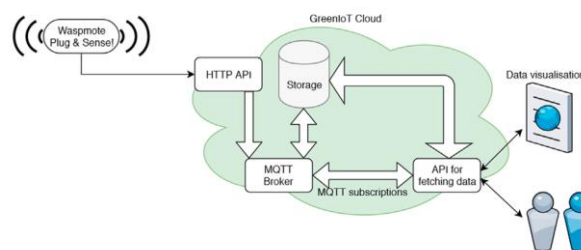
**Figura 6.** Netatmo NHC-EC (izquierda) y Netatmo NWS01-EC (derecha)

Fuente: Amazon, 2021o.

### 2.2.3. Proyectos con nuevas soluciones

A nivel académico también hay proyectos que han buscado soluciones aplicadas a la contaminación atmosférica.

Uno de los proyectos más interesantes, está realizado en Suecia, se trata de una monitorización mediante sensores inalámbricos instalados en los vehículos de transporte público, este proyecto se realiza en la ciudad de Uppsala a unos 78 km al noroeste de Estocolmo, se trata de la segunda ciudad más poblada de Suecia. Los dispositivos incluían sondas de sensor para mediciones de NO<sub>2</sub>, CO, temperatura, humedad y presión ambiental. También incluía una batería de litio recargable de 6600mAh junto a un panel solar para recargar el dispositivo. Con conexión a datos 4G y una antena GPS para determinar las distintas ubicaciones. El dispositivo aprovecha las rutas del bus con el fin de cubrir un área más grande de la ciudad. A través de la red 4G transmitía los datos a la API HTTP creando un JSON que se envía al MQTT bróker para crear subscripciones de los datos y los envía a MONGODB. En la Figura 7 se observa el esquema de comunicación y en la Figura 8 el dispositivo:



**Figura 7.** Esquema de la comunicación para la solución de medición a partir del transporte público

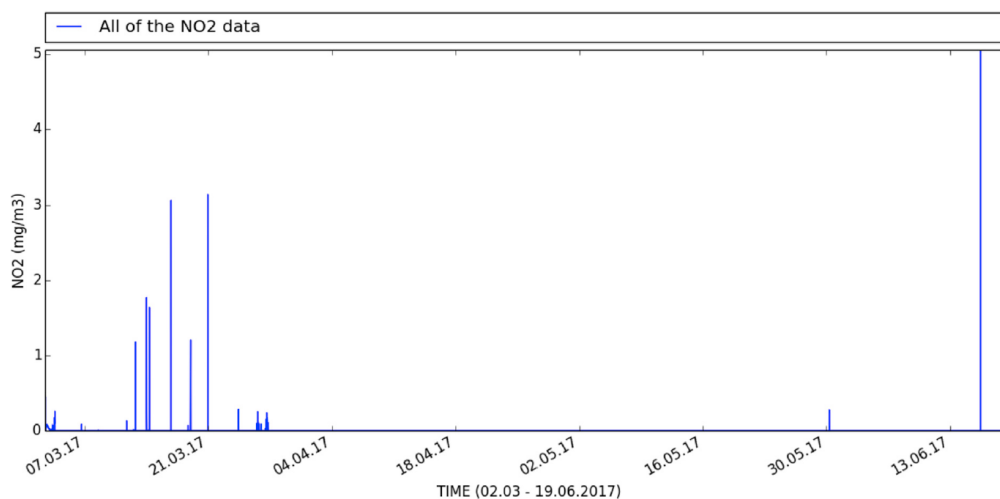
Fuente: Kaivonen et al., 2020



**Figura 8.** Dispositivo del proyecto de Suecia para la medición de contaminación

Fuente: Kaivonen et al., 2020

Una vez finalizado el estudio se observó que el  $\text{NO}_2$  normalmente no se midió bien debido a su baja sensibilidad, en la Figura 9 se observa como demasiadas veces dio valores de 0. Los datos se enviaron con red móvil 4G, con una tasa de éxito 70% sin reenvío. Y una tasa del 100% en el almacenamiento y retransmisión implementada. (Kaivonen et al., 2020).



**Figura 9.** Gráfica monitorización del  $\text{NO}_2$  del dispositivo

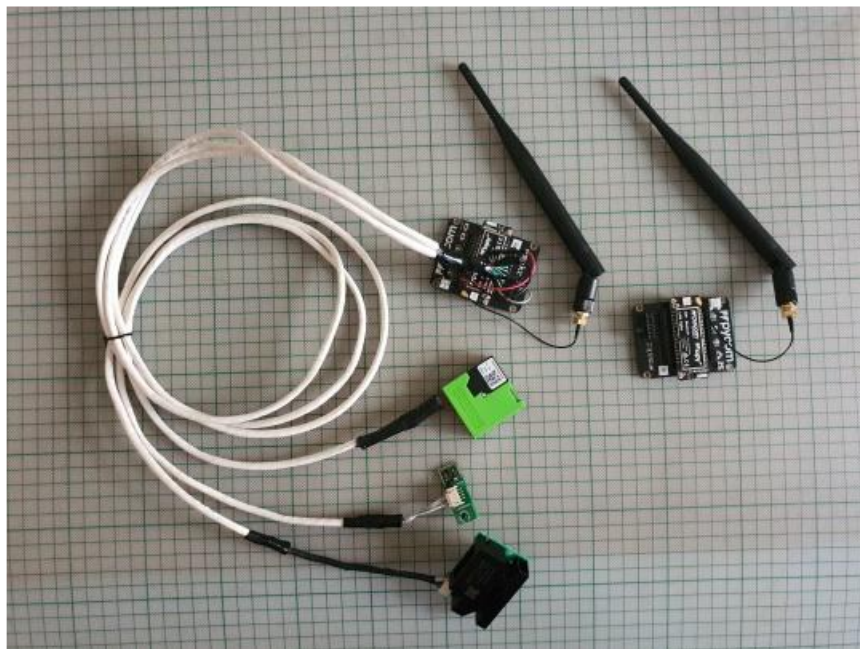
Fuente: Kaivonen et al., 2020

Entre otros proyectos de interés se puede destacar el de (Pelandarinos et al., 2021) Early Warning Systems for COVID-19 Infections Based on Low-Cost Indoor Air-Quality Sensors and LPWANS



En este proyecto se diseña un dispositivo que sea capaz de alertar infecciones por COVID-19 mediante la medición de la calidad del aire de interiores. Este proyecto mediante el uso de sensores PM, presión atmosférica, temperatura, humedad, COV. La implementación de conectividad es mediante conexión LoRaWAN. La red se centra en la detección de partículas PM en interiores, con la finalidad de disminuir COVID19, riesgo de infecciones por COVID 19, otras infecciones pulmonares, entre otras. Con mediciones fiables. (Pelandarinos et al., 2021)

En la Figura 10, se observa el hardware de este dispositivo.



**Figura 10.** *Dispositivo de medición de calidad del aire en interiores LoRaWAN*

Fuente: Pelandarinos et al., 2021

## 2.3. Tecnologías relacionadas con el proyecto

### 2.3.1. Habilitadores de la Industria 4.0

El proyecto se ha centrado en los siguientes habilitadores de industria 4.0:

- IoT (Internet of Things)
- Big Data
- Sensores y protocolo de comunicación
- Ciberseguridad

### 2.3.1.1. IoT

El concepto de Internet de las cosas o IoT (Internet of Things) nace en 1999 por Kevin Ashton, uno de los fundadores del laboratorio de reconocimiento automático de Massachusetts Institute of Technology. Se trata de una red de dispositivos cotidianos conectados a Internet. Este dispositivo puede tener integrado sensores que ayuden a la recopilación de datos de interés y transmitirlos a la red. Esto ayuda al control de estados internos, externos y la toma de decisiones (Radouan Ait Mouha, 2021)

Los elementos principales de estos sistemas están formados por:

- The Thing: Cualquier objeto o cosa, habitualmente incluye sensores o actuadores.
- Gateway: Encargado de comunicar los dispositivos entre sí, en protocolos
- Cloud Computing: Adaptación de capacidades informáticas a un sistema que utiliza los servicios en la nube, eso permite procesar datos de millones de dispositivos, usuarios y aplicaciones
- Analytics: Análisis de cantidades masivas de datos de dispositivos IoT, predecir patrones, tendencias y clave en la toma de decisiones
- User interface: La facilidad de uso de un dispositivo para los usuarios finales

El IoT ha presentado una gran popularidad rápidamente llegando a los 40 millones en 2020 y con unas expectativas de 75 millones en 2025.

Este crecimiento ha aumentado a la contribución de unos puntos claves: Entre los que se puede acentuar un posicionamiento y diversificación de la localización, el gran desarrollo de capacidad de Cloud Computing, los sistemas de aprovechar cantidades masivas de datos y la eficiencia y precisión de estos sistemas. (Chakroun et al., 2022)

Aunque también se debe ser consciente que hay aspectos que pueden presentar problemas o ser un desafío, es el caso de la seguridad, privacidad, interoperabilidad y normas, leyes y derechos

Entre las aplicaciones principales de IoT se puede señalar diferentes áreas: industria, transporte, energía, venta al por menor, smartcities, sanidad, cadena de suministros, agricultura y edificios inteligentes. (Radouan Ait Mouha, 2021)



### 2.3.1.2. Big Data

En concreto Big Data se trata de una cantidad masiva de datos, incapaz de almacenar, procesar o analizar con las formas habituales. Para la gestión de Big Data es necesario tener en cuenta los 5 parámetros, conocido como 5V: Volumen, Variedad, Velocidad, Valor y Veracidad.

El análisis Big Data hace referencia al proceso de análisis de grandes cantidades de datos y mediante diferentes técnicas permitirán identificar patrones, ver correlaciones o predecir tendencias que ayudan en la toma de decisiones, más rápidas y mejores o incluso en la reducción de costes. El proceso de análisis se basa en realizar inicialmente una recolección de datos de varias fuentes, hacer un análisis inicial de la información para organizar y segmentar, filtrar para asegurar su calidad, y entonces analizar con los softwares específicos de análisis.

El análisis Big Data ayuda a las empresas a aprovechar al máximo sus datos, consiguiendo mejores decisiones comerciales, operaciones más efectivas, reducción de costes, entre otras, todo esto generando mayores beneficios (Smaya, 2022).

### 2.3.1.3. Sensores y protocolos de comunicación

Los sensores y actuadores son componentes clave para IoT, ayudando en la recolección de datos e interacción con el sistema. Generalmente los datos se procesan y mediante comandos automatizados se activan actuadores. Entre los sensores más utilizados se puede distinguir sensores de temperatura, humedad, luz, acelerómetro, GPS, Giroscopio, magnetómetro, micrófono, sensor de pantalla táctil, sensor huellas dactilares, entre otros. Existen sensores físicos, mecánicos y químicos (Radouan Ait Mouha, 2021)

En la actualidad hay una gran cantidad de protocolos IoT entre ellos se puede destacar Constrained Application Protocol (CoAP), Advanced Message Queuing Protocol(AMQP), MQTT, Extensible Messaging and Presence Protocol (XMPP), WebSocket. (Kwizera et al., 2021)

### 2.3.1.4. Ciberseguridad

El concepto ciberseguridad hace referencia a la protección de la infraestructura ciber física, dando especial énfasis a la protección, privacidad y la confianza de información contenida o su circulación en las redes. En concreto las arquitecturas ciber-físicas tienen que cumplir las siguientes características de seguridad:

- Autenticidad: Solo pueden acceder al sistema los usuarios autorizados.

- Autorización: Limitación de privilegios o accesos a ciertos recursos.
- Confidencialidad: Proteger a la información frente a los intrusos.
- Integridad: La información no debe ser manipulada.
- Disponibilidad y continuidad: Garantizar un sistema sin fallos ni interrupciones operativas.

Por ese motivo es muy importante desarrollar procedimientos de seguridad, mediante estándares, protocolos, métodos, reglas o herramientas con el fin de minimizar los riesgos ante un ataque o amenaza. (Leloglu, 2017) (AIDosari, 2017)

### 2.3.2. Hardware

#### 2.3.2.1. Microcontrolador

Actualmente hay variedad de microcontroladores de bajo coste para la implementación de sensores. Los microcontroladores Espressif y Arduino permiten conectar distintos sensores y un funcionamiento rápido y eficaz. Este proyecto analiza el ESP32. Este microcontrolador permite una conexión a WIFI para poder enviar los diferentes datos recibidos por los sensores.

La Tabla 5 muestra una comparación de los microcontroladores.

**Tabla 5.** Comparación Microcontroladores

Sensor	Positivo	Negativo
<b>ESP32</b>	Menor precio Menor tamaño	Menor control Menor tratamiento de información
<b>ESP8266</b>	Menor precio Menor tamaño	Solo dispone de una entrada analógica Menor control Menor tratamiento de los datos
<b>Raspberry Pi</b>	Mayor Control Mejor control de la información Más funcionalidades	Mayor tamaño Mayor precio Necesidad de mayor energía

Fuente: Elaboración propia (Espressif), 2021n

### 2.3.2.2. Sensores

En la actualidad hay una gran variedad de sensores capaces de detectar los gases o partículas que ponen en riesgo la salud. En la Tabla 6 se muestra un listado con los sensores actualmente en el mercado para detectar contaminantes atmosféricos

**Tabla 6.** Sensores actualmente en el mercado para detectar contaminantes atmosféricos

<b>Sensor</b>	<b>Función</b>	<b>Positivo</b>	<b>Negativo</b>
<b>MQ135</b>	Medición de calidad del aire	Coste bajo	Calidad normal
<b>CCS811</b>	Medición de CO <sub>2</sub>	Coste bajo	Calidad normal
<b>MQ131</b>	Medición de O <sub>3</sub>	Coste bajo	Calidad normal
<b>MQ7</b>	Medición de CO	Coste bajo	Calidad normal
<b>ZMOD4510</b>	Medición NO <sub>2</sub> y O <sub>3</sub>	Buena calidad del sensor	Coste demasiado elevado
<b>B5WLD01011</b>	Detección de partículas 0,5 µm	Buena calidad del sensor	Coste demasiado elevado
<b>MiCS-6814</b>	Mediciones CO, NH <sub>3</sub> y NO <sub>2</sub>	Coste bajo Detecta distintos gases con un sensor	Calidad normal
<b>Alphasense CO<sub>2</sub></b>	Mediciones CO <sub>2</sub>	Buena calidad del sensor	Coste demasiado elevado
<b>Alphasense NO<sub>2</sub></b>	Mediciones CO <sub>2</sub>	Buena calidad del sensor	Coste demasiado elevado
<b>Alphasense O<sub>3</sub></b>	Mediciones O <sub>3</sub>	Buena calidad del sensor	Coste demasiado elevado

Fuente: Elaboración propia (Amazon 2021n)

### 2.3.3. Protocolo y comunicación

#### 2.3.3.1. MQTT

MQTT (Message Queue Server Telemetry Transport) es un protocolo basado en la suscripción y publicación, se trata de un protocolo de mensajería ligero de mensajería asíncrona, en base de comunicación TCP/IP y comunicación M2M (máquina a máquina), es muy utilizado para sistemas IoT, principalmente ya que puede ofrecer servicio a varios clientes, teniendo una estructura de estrella que permite conectar muchos clientes al Broker.

- El **bróker** es el servidor central que recibe los mensajes del cliente y los redirige a los clientes de destino. Responsable de filtrar y distribuir la información.
- El **cliente** sería cualquier dispositivo que interactúa con el bróker y que tenga conexión a internet, puede tratarse de publicadores o suscriptores. Los publicadores envían mensajes al Broker y los suscriptores reciben la información del Broker. Esta información también se llama Payload. El Payload son los datos transmitidos que constituyen el mensaje real.

La arquitectura del protocolo MQTT es de estrella, esto permite conectar muchos dispositivos a un mismo bróker. En la Figura 11 se puede observar la arquitectura del protocolo MQTT.

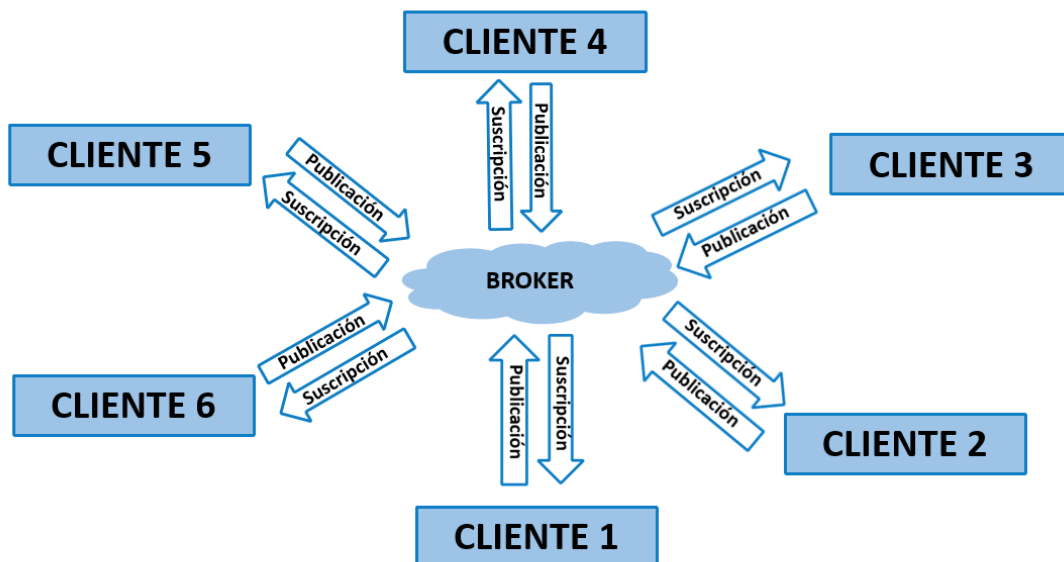


Figura 11. Arquitectura de estrella MQTT

Fuente: Elaboración propia

También incluye niveles de Calidad de Servicio (QoS) para garantizar que los datos se han enviado:

- QoS-0: Una vez como máximo, este nivel es el más bajo y hace que la entrega sea la más rápida, aunque no garantiza la entrega, se utiliza generalmente en conexiones estables en las que no importa si ocasionalmente se pierden algunos mensajes. Su uso es adecuado cuando se envía mensajes en intervalos de tiempo muy reducidos.
- QoS-1: Una vez como mínimo. En este nivel se garantiza que el mensaje se haya publicado al Broker pero se pueden recibir duplicados.
- QoS-2: Exactamente una vez. Este es el nivel más alto que garantiza que el mensaje se ha recibido exactamente una vez. Es el más seguro y confiable, pero a la vez el más lento. (Kwizera et al., 2021)

#### 2.3.3.2. Seguridad IoT

La seguridad IoT se distingue en 4 capas: Capa de percepción, Capa de red, Capa de soporte y Capa de aplicación

La seguridad en la capa de percepción, el primer paso y más importante es garantizar que solo las personas autorizadas puedan tener acceso a esos datos confidenciales, por eso es necesario tener unos requisitos de autenticación y autorización. En segundo lugar el proceso criptográfico de los datos que ha recibido el sensor, se puede distinguir el cifrado simétrico, cifrado asimétrico y acuerdo de clave asimétrica Hash, criptografía de curva elíptica ECC, la criptografía asimétrica adecuado para hardware con recursos limitados (Leloglu, 2017).

La seguridad en la capa de red, a nivel inalámbrico existe SSL/TLS (Secure Sockets Layers/ Transport Layer Security) está desarrollado con el objetivo de cifrar el enlace en la capa de transporte, el protocolo de seguridad IP (IPSec) para mantener la capa de red segura. El uso de PPSK (Private Pre-shared Key) seguridad a cada dispositivo conectado a la red, proporciona claves únicas, contraseñas seguras gestión de contraseñas y cambios periódicos de contraseña. Esto ayuda entre los puntos de seguridad de Autenticidad, Confidencialidad e Integridad. (Leloglu, 2017)

Finalmente, para la capa de soporte y aplicación también es adecuado el uso de autenticación, control de acceso, OPS, firewall, antivirus, verificación de datos y encriptación de datos.

#### 2.3.4. Herramientas para la realización del proyecto

##### 2.3.4.1. EMQX

EMQX se trata de un Broker MQTT de código abierto, muy escalable y con un alto rendimiento, permite conectar más de 100M dispositivos IoT. Capaz de mover y procesar millones de mensajes MQTT por segundo. (EMQX, 2022)

##### 2.3.4.2. Docker

Docker es una herramienta que permite controlar y gestionar contenedores. Los contenedores son una imagen de software ligero, independiente y ejecutable. Permite la ejecución de todo tipo de aplicaciones, múltiples lenguajes, marcos y arquitecturas.

Entre las características de Docker y sus contenedores son:

- Estándar: Gracias al uso de contenedores, es un sistema totalmente estándar.
- Peso ligero: Los contenedores comparten Kernel con el sistema operativo de la máquina por lo que no requiere un sistema operativo para la aplicación, permitiendo una reducción del uso de recursos del servidor.
- Seguridad: Las aplicaciones presentan más seguridad mediante el uso de contenedores y a parte Docker también proporciona capacidad de seguridad, permitiendo un sistema con más aislamiento y sólido. (Ratliff, 2022)

##### 2.3.4.3. InfluxDB

InfluxDB se trata de una plataforma de bases de datos muy adecuada para aplicaciones con series temporales, se trata de una serie de datos enviados en un momento concreto, donde se registra el momento de entrada de los datos y esto permitirá poder ordenar

cronológicamente el conjunto de datos. Este tipo de base de datos se adapta perfectamente a las necesidades de IoT. (InfluxData, 2021)

#### 2.3.4.4. Dashboard, Grafana y Android Studio

Dashboard es un panel de control, herramienta de gestión de la información que permite monitorizar, gestionar, analizar y mostrar datos de interés.

Grafana es una herramienta de código abierto capaz de crear paneles que permiten consultar y visualizar los datos, junto a la generación de alertas, sin la necesidad de importar los datos de donde se almacenan (Grafana Labs, 2022).

Android Studio se trata de un entorno robusto de desarrollo integrado de aplicaciones Android, construcción, diseño, maquetación, funcionalidad, entre otros. Permitiendo emular el dispositivo mientras se realiza el proyecto o se realizan modificaciones. (Funciones de Android Studio, 2022)

## 2.4. Conclusiones sobre el estado del arte

Con lo mencionado anteriormente, se ha podido observar que en los últimos años se ha trabajado en las soluciones enfocadas en la contaminación atmosférica, aunque al mismo tiempo presentaban unas ciertas debilidades.

A continuación, detallo las conclusiones obtenidas sobre el estado del arte:

- Los objetivos de la OMS, junto a otras organizaciones con la recolección de datos con dispositivos de control de la contaminación van más enfocados en intentar saber los niveles de contaminación en zonas amplias, esto provoca que no sea preciso para los usuarios. Uno de los principales problemas es la utilización de medición de concentraciones de contaminantes atmosféricos en lugares fijos, el principal problema que plantea es que en el caso de grandes ciudades no tiene una cobertura de monitorización adecuada, no es lo mismo medir el contaminante de una zona donde apenas pasan vehículos que otra zona donde el tráfico es muy intenso.
- Entre los dispositivos que hay actualmente en el mercado se puede distinguir los de coste muy elevado, más aplicado a nivel industrial, con un coste muy superior, por lo que no están al alcance del público en general. Las mediciones son muy precisas, sin

embargo, la monitorización y control deben hacerse desde el propio dispositivo u ordenador por lo que no permite una gestión de los datos adecuada para un usuario particular.

- Los dispositivos actuales de bajo coste detectan una poca variedad de gases, en muchos de los casos dan más importancia a las mediciones meteorológicas que a los gases contaminantes. Algunos incluso muestran los datos en una pantalla LCD, cosa que al usuario no le es cómodo, cada vez se busca más tener el control y seguimiento desde su propio teléfono móvil y recibir avisos directamente en el mismo.
- En los proyectos desarrollados hasta el momento se ofrecen soluciones, pero no aplicadas a la salud, al contrario, están más relacionadas con la medición de los gases. También se debe indicar que muchos proyectos no miden la gran mayoría de gases que dañan la salud, y se centran solo en partículas PM<sub>2.5</sub>, PM<sub>10</sub> y CO<sub>2</sub>. Otro inconveniente es que muchas soluciones solo tienen en cuenta interiores o rutas predefinidas, cuando lo que se buscaría son dispositivos que el individuo pueda llevar con él para así tener un seguimiento personalizado.

En conclusión, los dispositivos actuales pueden resultar muy útiles, pero no es lo óptimo para el usuario. El usuario no busca soluciones genéricas, busca soluciones personalizadas y con fácil acceso a la monitorización y control desde cualquier dispositivo móvil (smartphone o tablet), para poder ver la contaminación atmosférica de todas las zonas por las que transita. También es importante recalcar que cumpla con las recomendaciones que indica la OMS, en los proyectos mencionados se miden las concentraciones, pero no se personalizan los datos ni se analiza el tiempo al que se ha estado expuesto a un contaminante atmosférico. Como se ha indicado anteriormente tanto la concentración como el tiempo de exposición son factores muy a tener en cuenta. También hay que mencionar que en la actualidad hay una gran cantidad de sensores de bajo coste en el mercado, junto a otras herramientas de tecnología 4.0, esto permite ser capaces de diseñar, desarrollar e implementar una nueva solución.

La solución planteada incluye un sistema mucho más enfocado en la salud de las personas, para un seguimiento por los lugares donde se transita, monitorizando, controlando y generando avisos personalizados de los contaminantes desde el propio teléfono móvil del usuario. Y con especial atención a los parámetros de la OMS tanto de concentración como al tiempo de exposición.



## 3. Descripción general de la contribución del TFE

### 3.1. Objetivos

#### **Objetivo general**

El objetivo general del presente trabajo es diseñar y desarrollar un sistema monitorización, seguimiento y control de los gases y partículas de contaminación atmosférica, enfocado en la salud de las personas.

#### **Objetivos específicos**

Para alcanzar el objetivo general se plantean los siguientes objetivos específicos:

- Diseñar un dispositivo de adquisición de datos de medición de los contaminantes atmosféricos, gases o partículas que afectan a la salud de las personas, tanto en exteriores como interiores.
- Implementar un sistema de procesamiento, transmisión y tratamiento de datos del dispositivo de adquisición de datos, cumpliendo los requisitos de seguridad.
- Desarrollar un sistema para que el usuario puede añadir información personal (nombre, edad, enfermedades, genero, ubicación, entre otros)
- Diseñar un sistema para la visualización y la monitorización de los datos obtenidos, mediante el desarrollo de una aplicación móvil.
- Generar avisos personalizados cuando los niveles de contaminación son superiores y pueden afectar la salud, teniendo en cuenta los parámetros indicados por la OMS.

### 3.2. Metodología del trabajo

La elección de la metodología es un punto importante para el desarrollo de un proyecto. Debido a este motivo es necesario conocer las principales características de cada metodología, conociendo bien las ventajas y desventajas. Así analizar cuál se adapta mejor al proyecto.

Entre las metodologías que cada día están consiguiendo más relevancia se puede distinguir:

- **Modelo cascada:** Se trata de un proceso lineal, donde todos los pasos tienen un orden secuencial

- SCRUM: Basada en Sprint cortos, pequeños ciclos donde diariamente se analiza el proyecto con demostraciones y análisis retrospectivos
- Kanban: Uso de elementos visuales para agilizar el enfoque, los flujos de trabajo y proceso.
- Programación extrema: Para la gestión de proyectos dinámicos en plazos muy ajustados, principalmente se basa en ciclos cortos, por lo que se generan muchas versiones del producto. Esta metodología normalmente se utiliza en software, ya que permite fácilmente actualizar e implementar nuevas versiones. (Asana, 2021)

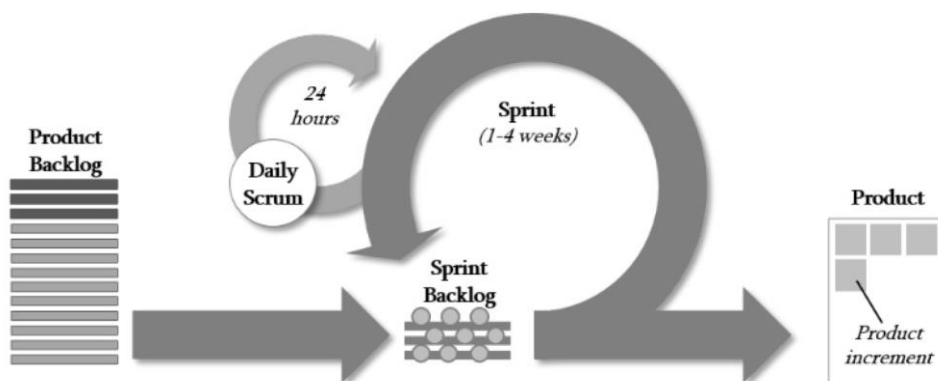
### 3.2.1. Elección y detalles de la metodología utilizada para el proyecto: SCRUM

La metodología utilizada para este proyecto es SCRUM, se trata de una metodología ágil que permite aportar al proyecto rapidez y flexibilidad.

A continuación, se detallan los elementos principales de la metodología SCRUM:

- Product Backlog: Se trata del conjunto de requisitos iniciales de producto. Sus características básicas.
- Sprint Backlog: La continuación del subproceso anterior a través de Sprints para realizar y revisar el desarrollo.
- Product Increment: Es el resultado final del producto. (Schuh et al., 2018); (Streule et al., 2016);

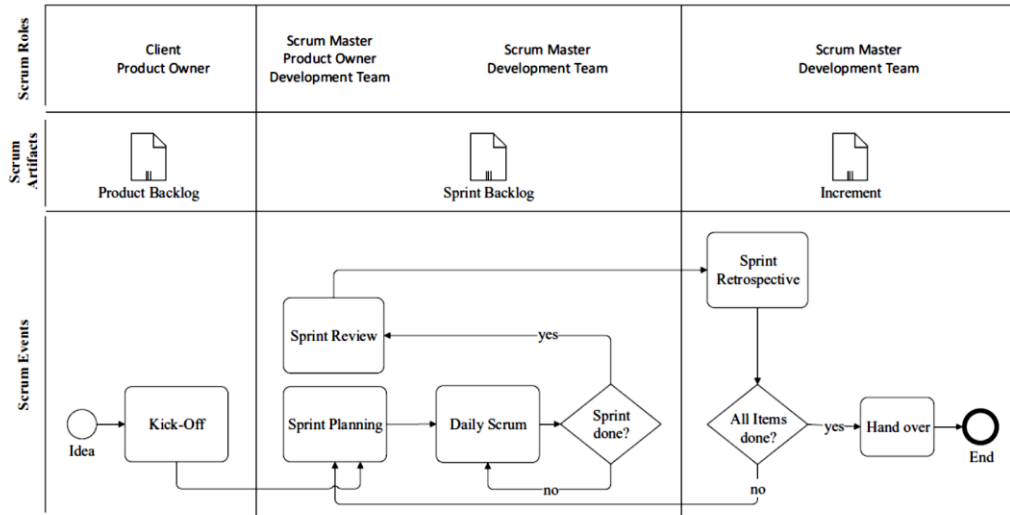
La Figura 12 muestra un esquema del funcionamiento de esta metodología:



**Figura 12.** Funcionamiento de la metodología SCRUM

Fuente: Schuh et al., 2018

En la Figura 13 se observa cuál es el marco típico para la metodología SCRUM, formado por roles, artefactos y eventos:



**Figura 13.** Marco típico de la metodología SCRUM

Fuente: Struele et al., 2016

Se ha decidido la utilización de la metodología SCRUM en este proyecto, principalmente por los Sprints, permitiendo solapar las diferentes fases del desarrollo. Esos Sprints siguen la siguiente estructura: planteamiento, desarrollo y revisión. Esto permite frecuentes revisiones de todos los desarrollos que se implementen y flexibilidad o adaptación al cambio en caso de ser necesario. Aunque se trate de un proyecto individual se tiene que ser crítico con los diferentes aspectos del proyecto por eso esta metodología busca una mejora continua del sistema, esto aporta un gran valor añadido ya que permite revisar y mejorar el sistema hasta conseguir que funcione de manera óptima. (Struele et al., 2016)

En particular en este tipo de proyectos son muy importantes las mejoras de forma progresiva de cada uno de los elementos del proyecto. En conclusión, se trata de una de las mejores metodologías en cuanto a la creación de un sistema.

### 3.2.2. Metodología y fases para el desarrollo del proyecto

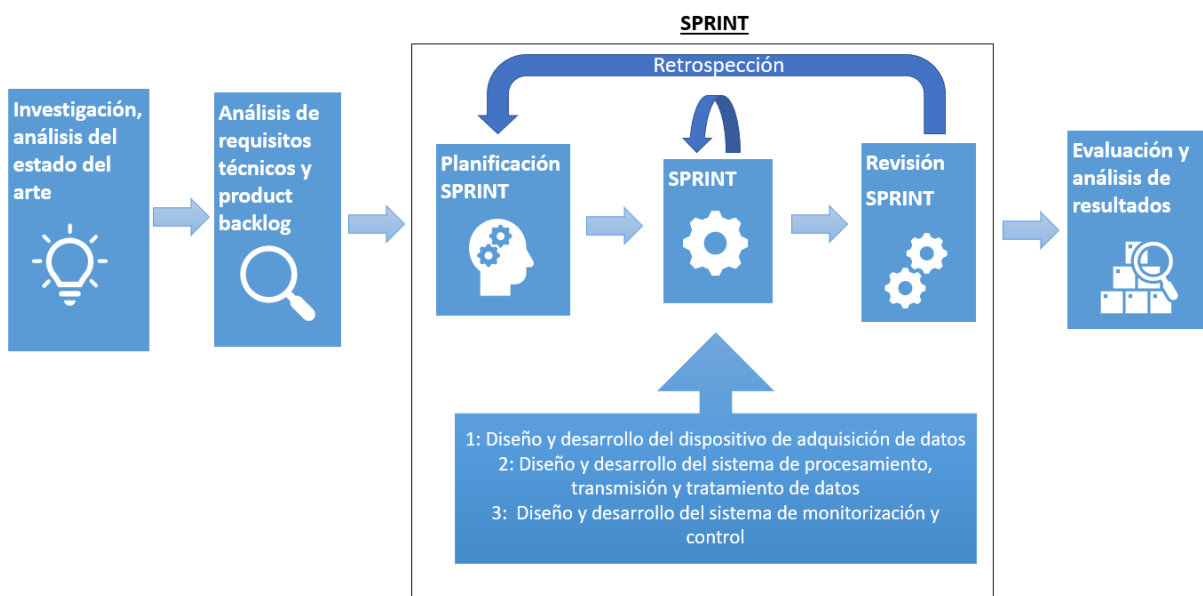
A continuación, se detallan las diferentes fases para el correcto desarrollo del proyecto usando la metodología SCRUM:

- **Investigación, análisis del estado del arte:** En esta primera etapa se analizan diferentes aspectos emblemáticos para el proyecto:
  - Gases contaminantes a los que se está expuesto
  - Limitaciones y recomendaciones de la exposición de gases
  - Efectos de la contaminación atmosférica sobre la salud de las personas
  - Soluciones actuales
  - Tecnologías relacionadas con el proyecto
- **Análisis de requisitos técnicos y product backlog:** En este se analiza la estructura de la solución, las diferentes partes que se deberá abarcar para poder diseñar y desarrollar la solución, tanto la creación del dispositivo de adquisición de datos, como el tratamiento de datos y su monitorización y control.
- **SPRINT:** Esta fase es la más importante para el desarrollo de la solución, estos estarán formados por 3 fases:
  - Planificación del SPRINT
  - El propio SPRINT para diseñar y desarrollar la solución
  - Revisión del SPRINT

Los SPRINTS están dentro de los tres desarrollos que se plantean en el proyecto solapándose para conseguir el objetivo planteado al final del proyecto:

- Diseño y desarrollo del dispositivo de adquisición de datos.
  - Diseño y desarrollo del sistema de procesamiento, transmisión y tratamiento de datos
  - Diseño y desarrollo del sistema de monitorización y control
- 
- **Evaluación y análisis de resultados:** En esta fase será necesario analizar si la solución diseñada cumple con todos los objetivos planteados y los requisitos establecidos. Y finalmente concluir si puede ser una solución viable y adecuada.

En la Figura 14 se detalla un diagrama resumen que describe las diferentes fases del proyecto:



**Figura 14.** Diagrama resumen de las fases del proyecto

Fuente: Elaboración propia

### 3.3. Descripción general de las partes o componentes de la propuesta

En este trabajo se realiza un sistema capaz de llevar un seguimiento de la contaminación atmosférica de donde transita el usuario, todo esto aplicado a la salud. Formado por un dispositivo para la adquisición de datos, el desarrollo de un sistema de procesamiento, transmisión y tratamiento de datos y finalmente el diseño y desarrollo de un sistema de monitorización y control que permitirá ver la contaminación a tiempo real, añadir los datos/perfil del usuario y recibir notificaciones o avisos personalizados.

#### **Alcance y limitaciones**

Mediante la realización de este trabajo se pretende tener un sistema de seguimiento y control de la contaminación atmosférica enfocado en la salud de las personas, que permita al usuario tener una monitorización, control y avisos personalizados, mediante una aplicación en su dispositivo móvil (smartphone o tablet). Para la creación de la solución será necesario un dispositivo de adquisición de datos de contaminación, formado por sensores conectados a un

microcontrolador que tendrá la función de Gateway, este transmitirá los datos a través de conexión a internet. Después es necesario una fase de procesamiento, transmisión y tratamiento de datos y finalmente diseñar y desarrollar un sistema de monitorización y control para el usuario.

Entre las posibles limitaciones:

- Problemas en la exactitud de las mediciones de algunos gases, debido al uso de sensores de bajo coste.
- Duración de la batería para poder llevar el dispositivo a todas las zonas por las que transita el usuario.
- Posibles limitaciones en el ámbito de tratamiento y seguridad de los datos

### **Tecnologías implicadas**

Entre las tecnologías utilizadas en este proyecto se puede señalar varias que forman parte de la Industria 4.0. Las principales tecnologías son dispositivos IoT, sensores y protocolos de comunicación, Big Data y ciberseguridad. También se debe remarcar que en el proyecto se ha decidido conseguir un sistema innovador utilizando las siguientes nuevas tecnologías:

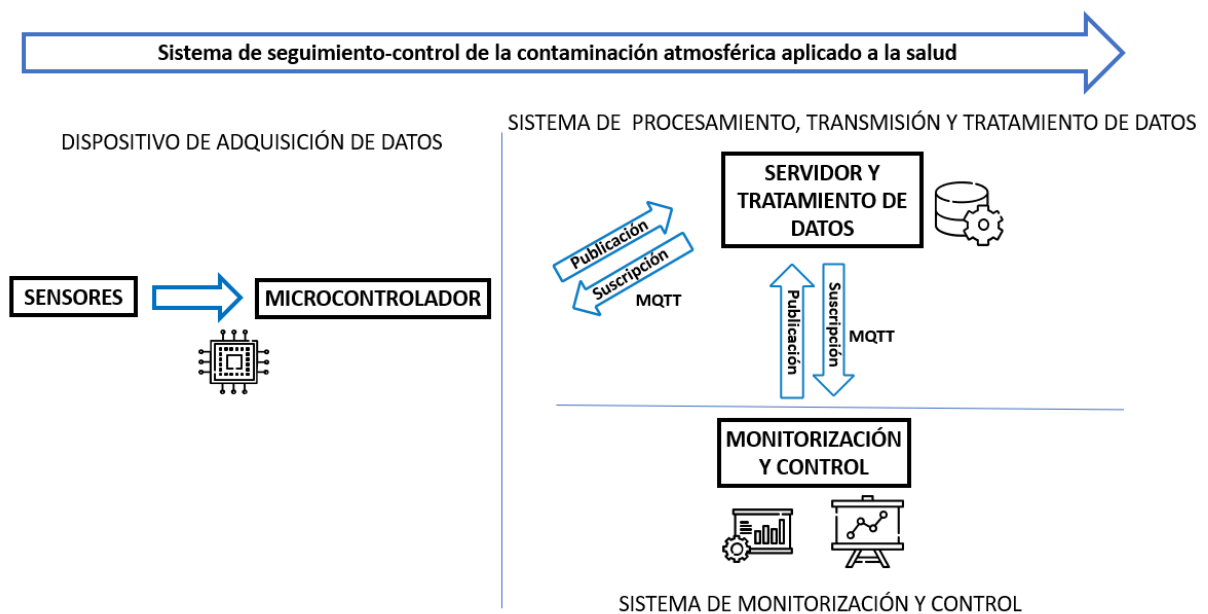
- IoT: Internet de las cosas, interconexión de objetos o dispositivos a una red.
- Sensores: Dispositivos capaces de detectar o extraer información de diferentes variables, tanto sean magnitudes físicas o químicas, transformándolas en variables eléctricas.
- Protocolos de comunicación: Protocolo de transmisión de datos o comunicación de dos o más entidades. Deben estar bien estructurados para una comunicación eficiente, segura, entre otras características.
- Big Data o análisis de datos: Análisis de grandes cantidades de datos.
- Ciberseguridad: Sistemas seguros basados en Autenticidad, Autorización, Confidencialidad e Integridad de los datos.

### **Arquitectura, componentes e integración de tecnologías**

En la descripción general el sistema estará formado por los siguientes elementos:

- **Dispositivo de adquisición de datos:** Esta parte del proyecto incluye los **sensores:** Encargados de la medición de los distintos gases y partículas que pueden dañar la salud y el **Gateway o microcontrolador** encargado de transmitir los datos de los sensores.
- **Sistema de procesamiento, transmisión y tratamiento de datos** encargado de gestionar y controlar los datos.
- **Sistema de monitorización y control:** Encargado de tener el control y monitorización del sistema.

En la Figura 15 se detalla un esquema de la arquitectura general del sistema:



**Figura 15. Arquitectura de funcionamiento del sistema**

Fuente: Elaboración propia

En el siguiente capítulo se detallarán cada uno de los elementos que forman parte del sistema.

### Resultados esperados

Se pretende generar un sistema fiable, personalizable, cómodo para el usuario y que aporte un valor, para tener en todo momento, tanto en interiores o exteriores la información de la contaminación atmosférica del aire que respira el usuario a tiempo real, para que el usuario pueda prevenir o mitigar los posibles riesgos por contaminación.

### Presupuesto y retorno esperado de la inversión

El presupuesto del proyecto incluye el coste de cada dispositivo de medición de 116,50 euros, formado por microcontrolador, sensores, cables, entre otros elementos. El sistema debe incluir el coste del dispositivo e implementación, costes administrativos, Seguridad Social (autónomos), Hacienda, patentes o modelo de utilidad.

Se busca que la solución sea low cost para los clientes de modo que sea asequible para gran parte de la población. En el siguiente capítulo, en el apartado 4.4, se detalla el presupuesto desglosado por cada uno de los elementos que forman la solución.

### Planificación general

El proyecto planificado según las siguientes fases:

1. Investigación y análisis del estado del arte
2. Análisis de requisitos técnicos y producto backlog
3. Desarrollos (Formado por constantes SPRINTS en cada uno de los desarrollos)
4. Evaluación y análisis de resultados

Y en la Figura 16 se muestra el cronograma/diagrama de Gantt del proyecto

SISTEMA DE SEGUIMIENTO-CONTROL DE LA CONTAMINACIÓN ATMOSFÉRICA APLICADO A LA SALUD	Mes 1	Mes 2	Mes 3	Mes 4	Mes 5	Mes 6	Mes 7
<b>Investigación y análisis del estado del arte</b>							
<b>Análisis de requisitos técnicos y product backlog</b>							
<b>SPRINTS:</b>							
1 Diseño y desarrollo del dispositivo de adquisición de datos							
2 Diseño y desarrollo del sistema de procesamiento, transmisión y tratamiento de datos							
3 Diseño y desarrollo del sistema de monitorización y control:							
<b>Evaluación y análisis de resultados</b>							

Figura 16. Cronograma/Diagrama de Gantt del proyecto

Fuente: Elaboración propia



## 4. Desarrollo específico de la contribución

### 4.1. Diseño y desarrollo del dispositivo de adquisición de datos

En este apartado se analizarán los elementos que se utilizan en el proyecto:

#### 4.1.1. Elección de microcontrolador y sensores

En este apartado se explicarán conceptos básicos aplicados en el proyecto como Microcontrolador y sensores que forman parte de la solución.

##### 4.1.1.1. Microcontrolador

Entre las características necesarias del microcontrolador se puede destacar que sea de pequeño tamaño a fin de permitir desplazarse con facilidad, además debe tener un bajo consumo energético, para funcionar durante el máximo tiempo posible sin necesidad de recarga.

A partir de la comparación de la Tabla 5 y de las características descritas, el Microcontrolador más adecuado para el desarrollo del proyecto es el ESP32.

Entre las ventajas principales que aporta son la fácil conexión USB, alimentación de 5V o 3,3 V, 18 entradas de Analógico a Digital (ADC). (Espressif, 2021).

En la Figura 17 se muestra el esquema del microcontrolador ESP32 DevkitC y sus GPIO (Pins de conexión de entrada y salida):

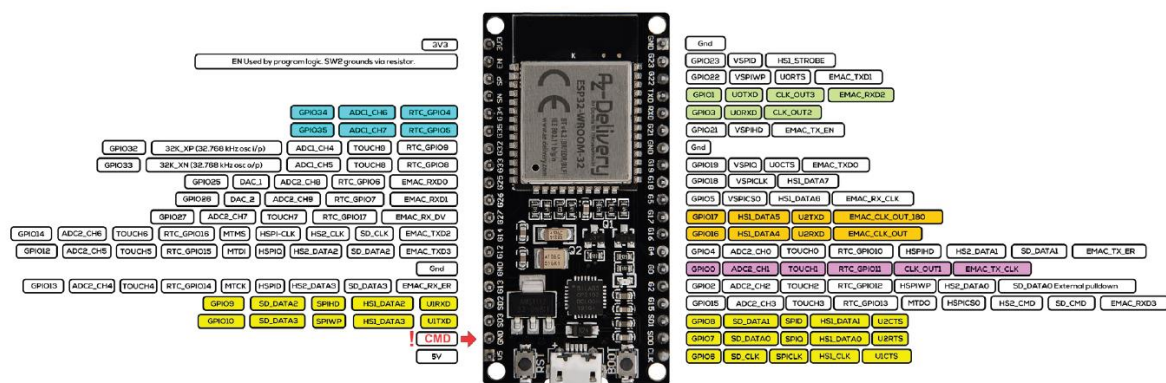


Figura 17. Esquema ESP-32 NodeMCU Developmentboard Ponout Diagram

Fuente: AZ-Delivery, 2020

#### 4.1.1.2. Sensores

A continuación, se detallan los distintos sensores que se utilizan en el proyecto:

- **MQ135:** Mediante este sensor de calidad de aire se podrá analizar el CO<sub>2</sub>.
- **MQ131:** Permite la detección de O<sub>3</sub>.
- **DHT22:** Permite la detección de temperatura y humedad.
- **MiCS-6814:** Este sensor es capaz de detectar tres gases el NO<sub>2</sub>, CO y NH<sub>3</sub>.
- **Waveshare Dust Sensor Detector Module with Sharp GP2Y1010AU0F:** Este sensor detecta las partículas que están en el aire hasta un tamaño de 0,8 µm, suficiente para medir las partículas PM<sub>10</sub> y PM<sub>2.5</sub>.

En la Figura 18 se muestra los sensores utilizados para la monitorización de la contaminación atmosférica



**Figura 18.** Sensores utilizados para el sistema de monitorización y seguimiento

Fuente: Amazon, 2021n

#### 4.1.1.3. Alimentación

Para la alimentación del dispositivo se ha analizado las distintas formas de alimentar el microcontrolador y los distintos sensores. La Tabla 7 muestra una comparación de los diferentes tipos de alimentación para saber cuál puede ser mejor para el proyecto.

**Tabla 7.** Comparación de baterías

Tipo de alimentación	Ventajas	Desventajas
<b>Batería 18650</b>	Coste no elevado Comodidad a la carga	Voltaje poco elevado, pero con componentes puede llegar a los 5V necesarios
<b>Pilas 1,5V</b>	Bajo coste	Demasiado poco suministro (es necesario 5V) Poco tiempo de suministro
<b>Banco de energía 5V</b>	Comodidad al cargar	Voltaje correcto Coste elevado

Fuente: Elaboración propia (Espressif y Amazon), 2021n

La mejor opción para nuestro dispositivo es la batería 18650, uno de los principales motivos es la necesidad de alimentación de sensores en un voltaje de 5 voltios. En la Figura 19 se muestra una imagen del modelo de batería junto al dispositivo que se utilizara para alimentar al ESP32 y los sensores.

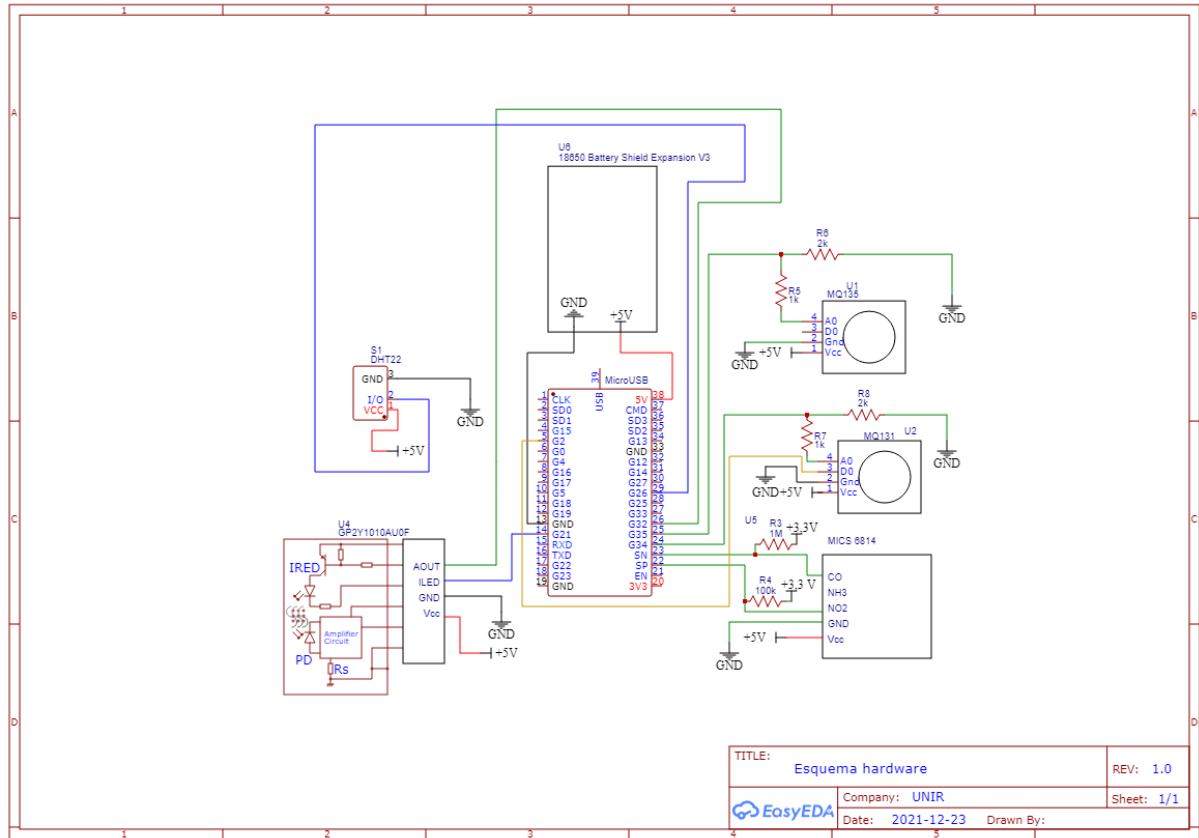


**Figura 19.** Alimentación y batería para sensores y microcontrolador

Fuente: Amazon, 2021n

#### 4.1.2. Esquema del dispositivo

En la Figura 20 se puede observar un esquema del hardware utilizado para la creación del dispositivo:



**Figura 20. Esquema hardware con EasyEDA**

Fuente: Elaboración propia con el Software EasyEDA

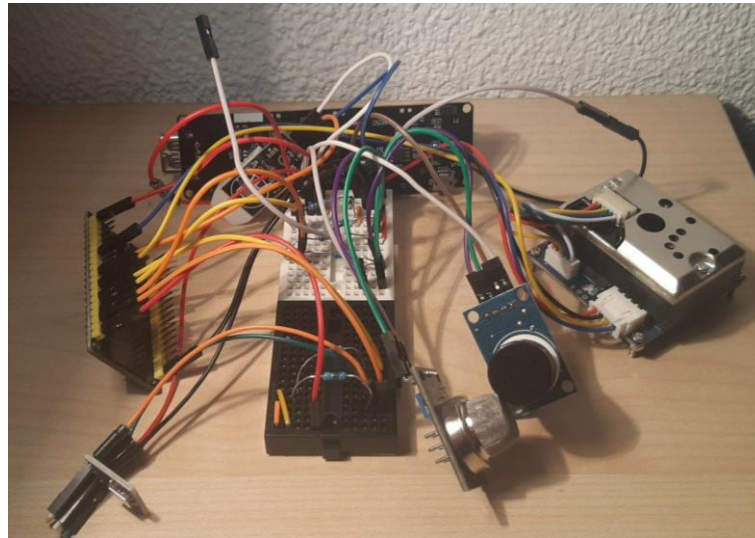
Como se puede observar ha sido necesario poner resistencias, para obtener buenos resultados en los sensores, en el caso de los sensores MQ135 y MQ131, se ha hecho un divisor de tensión con el fin de reducir el voltaje de 5V a 3,3V, ya que ese es el máximo valor de voltaje que pueden leer las entradas ADC del microcontrolador.

A continuación, se detalla la fórmula utilizada para el cálculo, y así saber que resistencias utilizar para el divisor de tensión:

$$V_{out} = \frac{R1}{R2 + R1} \cdot V \quad (\text{Ec. 1})$$

$$V_{out} = \frac{2\ k}{1\ k + 2\ k} \cdot 5 = 3,33\ V$$

Así mediante una resistencia de 1k y otra de 2k, se logra el voltaje adecuado de 3,3V para la lectura ADC de los sensores analógicos. A continuación, en la Figura 21 se muestra el dispositivo de adquisición de datos incluyendo todas las conexiones, sensores, microcontrolador, resistencias, batería y un protoboard para hacer más cómodas las conexiones del prototipo.



**Figura 21.** Hardware del dispositivo

Fuente: Elaboración propia

#### 4.1.3. Funcionamiento del dispositivo

El funcionamiento del dispositivo de adquisición de datos se puede resumir de la siguiente manera, conectando los sensores a los pins de entradas y salidas del microcontrolador, este último procesará y transmitirá la información mediante conexión a internet.

El sensor MQ135 se encarga de las mediciones de CO<sub>2</sub>, actualizando el valor de concentración en ppm cada minuto. Según se indica en la Tabla 3 la máxima exposición recomendada para la salud es de 1.500ppm por eso en caso de que sea superior se generará un aviso al dispositivo móvil, también se generarán avisos a niveles intermedios según lo recomendado en interior o exterior. Se puede diferenciar entre dos tipos de avisos según si es interior o exterior, en el caso de interior la alerta irá enfocada a abrir ventanas o ventilación. Junto a este sensor, a fin de poderlo calibrar de forma adecuada, se utiliza el sensor DHT22 realizando mediciones de temperatura y humedad, estas se van publicando y actualizando cada minuto.

En el caso del sensor MQ131 y MICS6814, para las mediciones de concentraciones de O<sub>3</sub>, CO y NO<sub>2</sub> con actualizaciones cada minuto de forma automática, se detectará y avisará cuando los valores sean superiores a los recomendados en la Tabla 1, también se han diferenciado los límites recomendados para cada usuario según sus patologías previas. Implementando contadores con el fin de tener un control del tiempo al que se ha estado expuesto a estos contaminantes, según los límites indicados en las Tablas 2 y 3. Finalmente, en el caso del sensor Waveshare Dust Sensor Detector Module with Sharp GP2Y1010AU0F para las partículas PM, indica la concentración de partículas PM<sub>2.5</sub> y genera avisos en caso de superar los límites recomendados según las características del usuario.

En la Figura 22 se observa un diagrama inicial que relaciona las variables para las decisiones y la acción o proceso que se realizará.

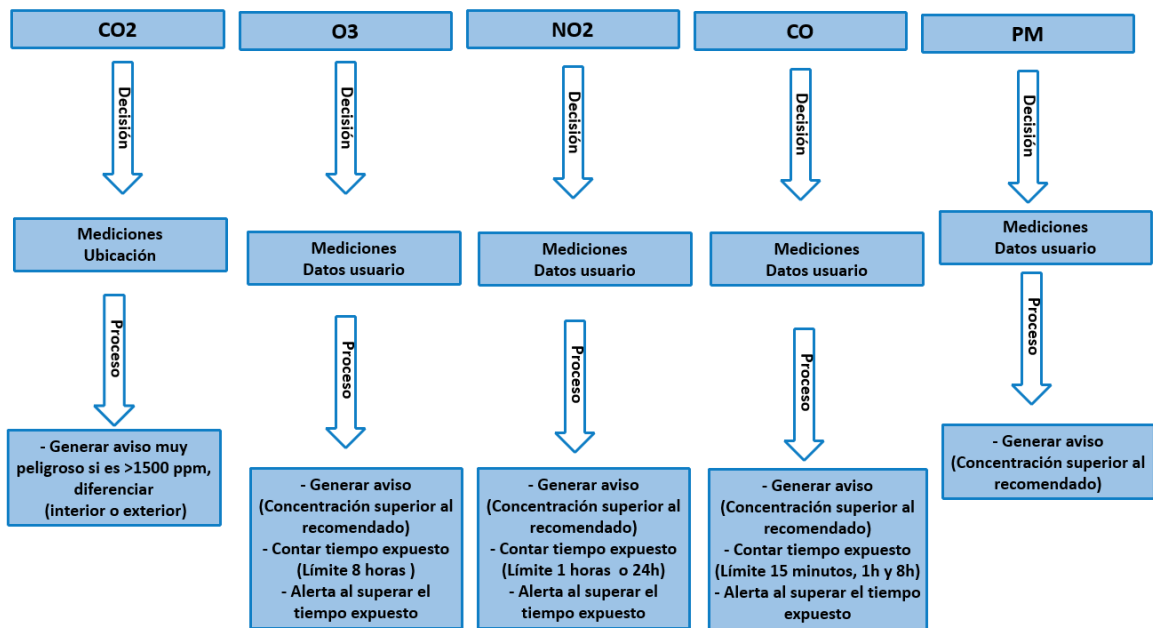
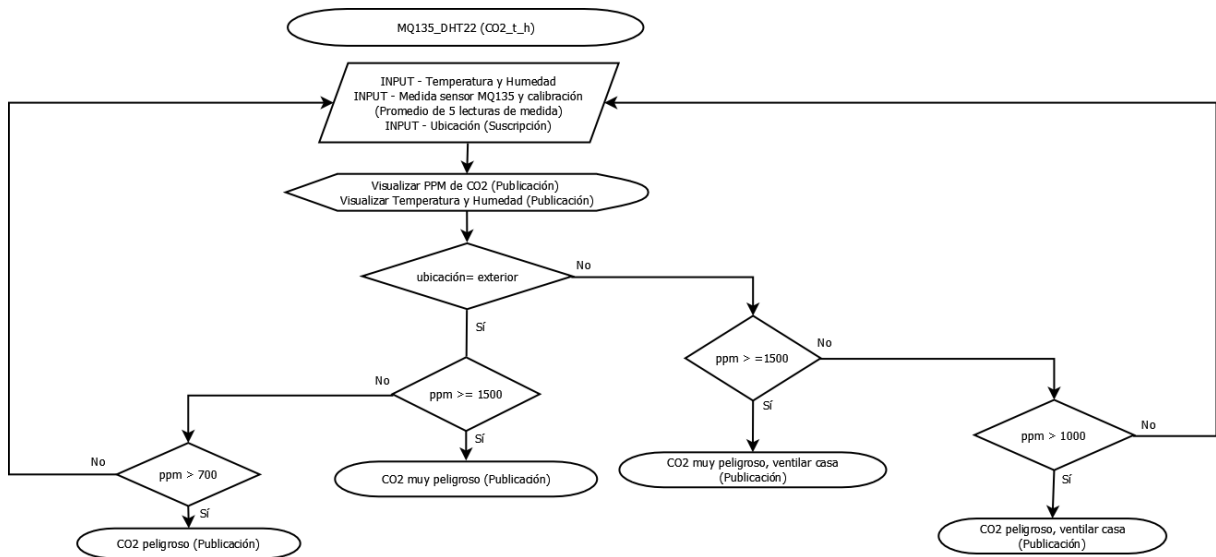


Figura 22. Diagrama inicial resumen del funcionamiento

Fuente: Elaboración propia

También se han realizado diagramas más específicos para definir el funcionamiento de cada uno de los sensores, en la Figura 23 se puede observar el diagrama de funcionamiento del sensor MQ-135 en el microcontrolador.



**Figura 23.** Diagrama funcionamiento del Sensor MQ135 en el microcontrolador

Fuente: Elaboración propia con Dia

El funcionamiento para el resto de los sensores se puede observar en el Anexo E.

## 4.2. Desarrollo del sistema de procesamiento, transmisión y tratamiento de datos

Para la correcta transmisión de los sensores es necesario un sistema para el procesamiento y transmisión de datos. Por ello es importante que el microcontrolador pueda transmitir los datos al Broker y este los gestiona para enviarlos a los distintos clientes. En el Anexo A se adjunta todo el código implementado para el correcto funcionamiento del sistema.

### 4.2.1. Protocolo de comunicación

El microcontrolador permite comunicar los sensores mediante conexión WIFI esto permite enviar los datos a través del dispositivo de adquisición de datos, sin la utilización de cables para la comunicación entre dispositivos.

El protocolo pensado para este proyecto es el MQTT (Message Queue Telemetry Transport). El protocolo MQTT es un protocolo de mensajería publicación/suscripción, asíncrono con una comunicación M2M, muy utilizado en sistemas IoT. Se trata de un protocolo de código abierto, ligero y flexible. Muy adecuado para el uso de sensores. Como se ha indicado anteriormente hay dos elementos fundamentales.

- El **bróker**: El servidor Cloud o EMQX, capaz de recibir los mensajes del cliente y los redirige a los clientes de destino.
- El **cliente**: El dispositivo de adquisición de datos de contaminación, la aplicación móvil u ordenadores de los usuarios, pudiendo ser cualquier dispositivo que interactúa con el bróker mediante conexión a internet, pueden ser publicadores y/o suscriptores.

En el proyecto ha sido necesario definir varios topics en distintos niveles:

- ID\_dispositivo/Contaminante/#: En el subnivel de este topic se encuentra la recepción de todas las mediciones de contaminantes del usuario asociado a un ID del dispositivo.
- ID\_dispositivo/Tiempo/#: En el subnivel de este topic se encuentra la recepción de todos los tiempos de exposición.
- ID\_dispositivo/Aviso/#: En el subnivel de este topic se encuentra la recepción de todos los avisos.

Se añade delante el ID\_dispositivo a fin de diferenciar usuario-dispositivo.

Y también el propio dispositivo de adquisición de datos está suscrito a topics en concreto la edad, enfermedades y ubicación del usuario para saber si está en interiores y exteriores. Para esta funcionalidad se ha utilizado el código de la Figura 24.



```

//Bucle lectura de los payload de los topics suscritos
void callback(char* topic, byte* payload, unsigned int length)
{
  String topicStr = topic; //Convertir a string para comparar topic
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.println(topicStr);
  char buff_p[length];
  if (topicStr == "595012/enfermedad/respiratoria") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_respiratoria_p = String(buff_p); //Extraer payload como string
    e_respiratoria = e_respiratoria_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/enfermedad/cardiovascular") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_cardiovascular_p = String(buff_p); //Extraer payload como string
    e_cardiovascular = e_cardiovascular_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/enfermedad/oncologicas") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_oncologicas_p = String(buff_p); //Extraer payload como string
    e_oncologicas = e_oncologicas_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/alergias") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String alergias_p = String(buff_p); //Extraer payload como string
    alergias = alergias_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/edad") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String edad_p = String(buff_p); //Extraer payload como string
    edad = edad_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/embarazo") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String embarazo_p = String(buff_p); //Extraer payload como string
    embarazo = embarazo_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/ubicacion") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar último payload del topic
    String ubicacion_c = String(buff_p); //Extraer payload como string
    ubicacion = ubicacion_c.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/tiempo/reiniciar") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String t_reinicio_p = String(buff_p); //Extraer payload como string
    t_reinicio = t_reinicio_p.toInt(); //Convertir string a entero
  }
}

```

Figura 24. Código para la lectura de valores de topics para el dispositivo

Fuente: Elaboración propia

#### 4.2.2. Broker Cloud

Una solución para el proyecto sería la implementación de un Broker Cloud, que permita la comunicación del dispositivo de adquisición de datos y las aplicaciones móviles para la monitorización y control. En la Figura 25 se puede observar la arquitectura de la solución con Servidor Cloud MyQttHub.

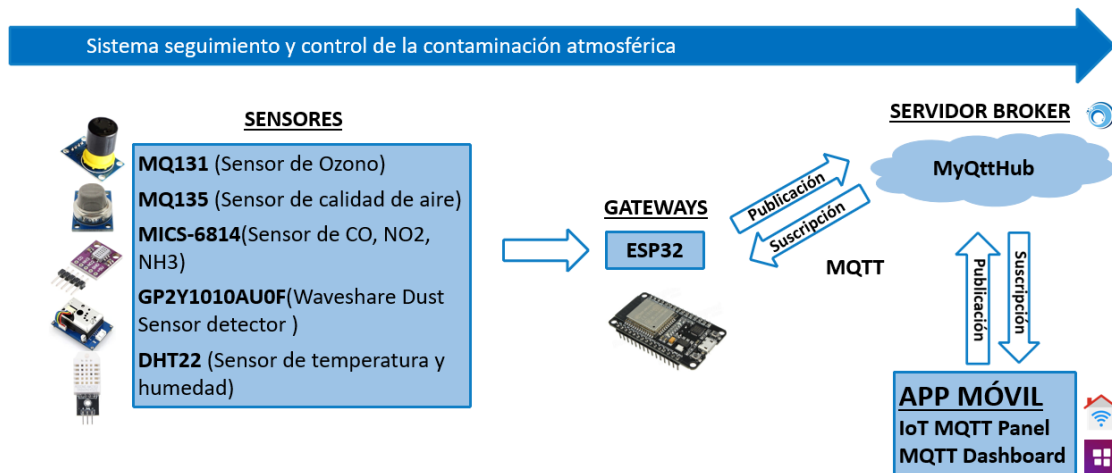


Figura 25. Arquitectura solución Servidor Cloud MyQttHub

Fuente: Elaboración propia

El Broker Cloud tiene ventajas a nivel energético, para no utilizar un dispositivo de mayor tamaño o un procesador de más alto nivel, también permite conectar desde cualquier ubicación y así cumplir con los objetivos de nuestro dispositivo. Uno de los mejores Broker online es MyQttHub (Figura 26), que ofrece unas características óptimas.

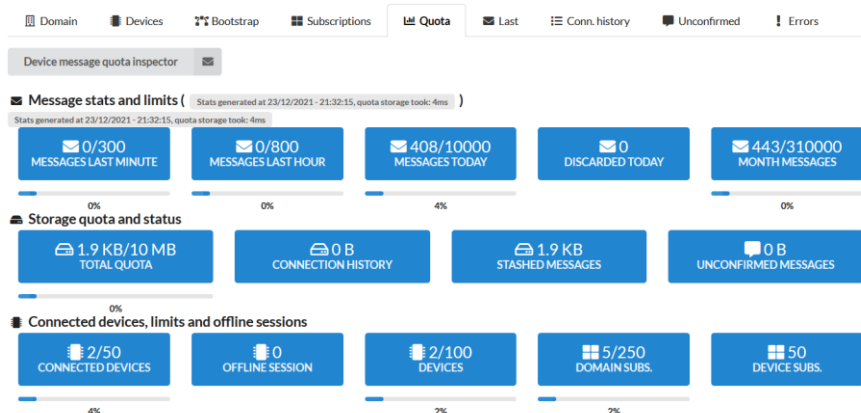
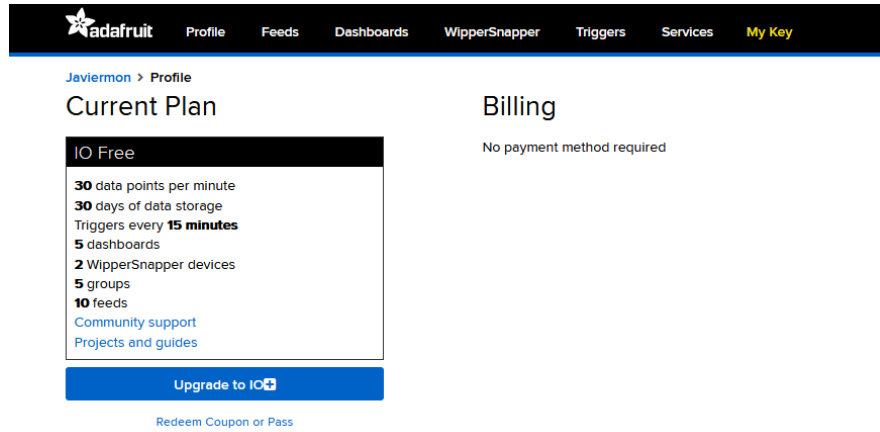


Figura 26. Características del Servidor gratuito de MyQttHub

Fuente: MyQttHub, 2021

Comparando con otras alternativas, esta ofrece unas características mejores en cantidad de suscripciones y dispositivos conectados.

La alternativa gratuita de Adafruit ofrece menos opciones en este aspecto y la comunicación sería menos adecuada para el tipo de proyecto que se está desarrollando: (Figura 27)



**Figura 27.** Características del Servidor gratuito de Adafruit

Fuente: Adafruit, 2021

La Tabla 8 muestra un resumen de la comparativa de los servidores gratuitos MQTT, MyQttHub y Adafruit.

**Tabla 8.** Comparación de características servidores gratuitos

Características	Dispositivos	Almacenamiento	Suscripciones	Mensajes Minutos	Mensajes hora	Mensajes día	Cifrado MQTT TLS	Precio
<b>MyQttHub</b>	50	10MB	50	300	800	10.000	Si	Gratis
<b>Adafruit</b>	2	30 días	10	30	1.800	43.200	Si	Gratis

Fuente: MyQttHub y Adafruit, 2021

Analizando las dos alternativas se ha considerado mejor la opción de MyQttHub ya que permite tener más hasta 50 suscripciones, necesarias debido a que el procesamiento de los sensores se realizará en el microcontrolador e informará a la aplicación mediante suscripciones en los topics.

#### 4.2.2.1. Realización de pruebas con el servidor

Para ver su funcionamiento, se muestra en la Figura 28 una captura del desarrollo y ver como registran los distintos mensajes que envían y como se asocian a un topic.

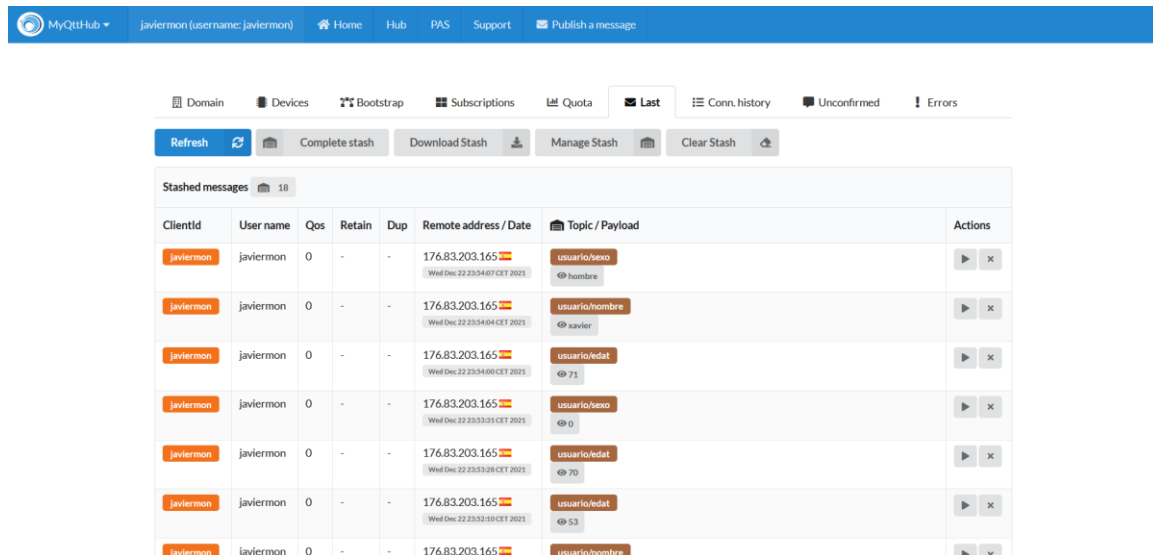


Figura 28. Broker MyQttHub recibiendo datos del cliente

Fuente: MyQttHub, 2021

Con el fin de un mejor tratamiento de datos, seguridad y posibilidad de configuración del servidor en el siguiente apartado se detalla el desarrollo a través de un sistema de servidor Broker propio.

#### 4.2.3. EMQX + INFLUXDB + GRAFANA

La solución de servidor EMQX permite un tratamiento de datos, tener un histórico de datos e incluir medidas de seguridad para la comunicación, entre otras funcionalidades. A continuación, en la Figura 29 se detalla el esquema general del proyecto, incluyendo la comunicación de los tres sistemas, dispositivos de adquisición de datos, sistema de procesamiento, transmisión y tratamiento de datos y sistema de monitorización y control.

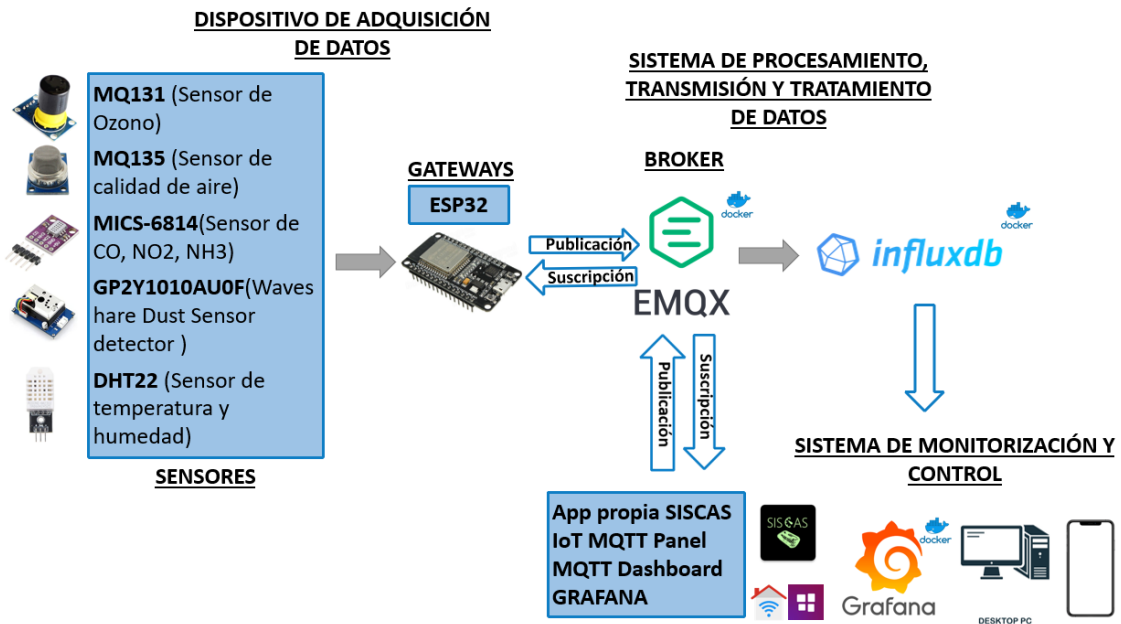


Figura 29. Arquitectura del proyecto

Fuente: Elaboración propia

Para aplicar la solución se ha utilizado Linux en concreto Ubuntu 20.04, se ha trabajado con Docker por las principales ventajas al ser estándar, muy ligero y seguro. Estas ventajas son claves para el desarrollo de la solución.

Para la creación del sistema de procesamiento, transmisión y tratamiento de datos se ha desarrollado un contenedor con EMQX y InfluxDB. En el Anexo C se adjunta el código de Docker para el desarrollo de contenedores EMQX, InfluxDB, Telegraf y Grafana.

#### 4.2.3.1. EMQX

Para la comunicación del dispositivo de adquisición de datos se ha pensado la utilización de EMQX como Broker por sus características de código abierto, estabilidad, rendimiento y sobre todo la posibilidad de modificar o implementar con otras funcionalidades. Para implementarlo es necesario activar el servicio con la imagen de EMQX, abrir los puertos necesarios para la comunicación, el puerto 18083 del Dashboard del EMQX de configuración, los puertos de comunicación 1883 y 8883 para los mensajes y crear los volúmenes de datos necesarios. En la Figura 30 se observa el código para crear el servicio EMQX.

```

version: "3"                                #Ultima version docker
services:                                    #Servicios que tendra el docker

  emqx:                                       #Nombre del servicio
    user: root                               #Debemos dar al usuario del servicio los permisos root
    image: emqx/emqx:5.0.4                   #Imagen de docker que utilizaremos en este caso EMQX

    environment:
      - EMQX_LISTENER__SSL__EXTERNAL__ACCEPTORS

    ports:                                    #Definir puerto que mapea y el puerto del contenedor
      - "18083:18083" #activar puerto 18083 Dashboard
      - "1883:1883"  #activar puerto 1883 publicación/suscripción
      - "8883:8883"  #activar puerto 8883 segura
      - "8083:8083"
      - "8084:8084"

    volumes:
      - ./data/emqx/data:/opt/emqx/data      #Directorio de volumen de datos de data
      - ./data/emqx/log:/opt/emqx/log
      - ./data/emqx/etc/certs:/opt/emqx/etc/certs

```

**Figura 30.** Docker-compose código servicio EMQX

Fuente: Elaboración propia con la Visual Studio Code

#### 4.2.3.2. INFLUXDB

Para el proceso de tratamiento de datos de contaminantes, una vez entran al Broker EMQX, se redirigirán a InfluxDB, se utiliza InfluxDB para la solución, ya que nuestros datos se asocian a un momento concreto por lo que esta base de datos es idónea para series temporales.

En la Figura 31 se observa el código para crear el servicio de bases de datos de InfluxDB.

```

influxdb:                                    #NOMBRE DEL SERVICIO
  user: root                                 #Dar permisos superusuario
  image: "influxdb:2.2.0-alpine"            #Imagen de docker Influxdb
  ports:                                     #Definir puerto mapeado y el puerto del contenedor
    - "8086:8086"                            #Puerto utilizado de Influxdb para conectar
  volumes:
    - ./data/influxdb/data:/var/lib/influxdb2 #Directorios de volumen de datos y configuracion de Influxdb
    - ./data/influxdb/config/etc/influxdb2
  environment:
    #Variables de entorno Influxdb
    - DOCKER_INFLUXDB_INIT_MODE=setup
    - DOCKER_INFLUXDB_INIT_USERNAME=*****
    - DOCKER_INFLUXDB_INIT_PASSWORD=*****
    - DOCKER_INFLUXDB_INIT_ORG=javiermontfe   #Organizacion para almacenar los datos
    - DOCKER_INFLUXDB_INIT_BUCKET=datostfe   #Bucket donde se almacenan los datos
    - DOCKER_INFLUXDB_INIT_RETENTION=1w
    - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=javiermontfe-datostfe-auth-token #Token utilizado para conectar los diferentes servicios

```

**Figura 31.** Docker-compose código servicio InfluxDB

Fuente: Elaboración propia con la Visual Studio Code

Para acceder a los datos de EMQX se utiliza la herramienta Telegraf un agente que permite la recopilación de datos y métricas, sirve de enlace para EMQX y InfluxDB, en la Figura 32 se muestra el código.

```

telegrafoutput:                                #NOMBRE DEL SERVICIO
user: root                                     #Dar permisos superusuario
image: "telegraf:1.22.4"                       #Imagen de docker Telegraf
deploy:
  restart_policy:
    condition: on-failure                       #Reiniciar servicio en caso de fallo, para iniciar posteriormente del EMQX
    delay: 10s
    max_attempts: 20                            #20 Intentos para la comunicacion cada 10 segundos
volumes:
  - ./data/telegrafoutput/telegraf.conf:/etc/telegraf/telegraf.conf #Directorios de volumen de configuracion datos output (Salida) EMQX

```

**Figura 32.** Docker-compose código servicio Telegraf

Fuente: Elaboración propia con la Visual Studio Code

Ha sido necesario crear un documento de configuración de Telegraf para la comunicación de EMQX y InfluxDB, permitiendo configurar el punto de entrada de datos y el punto de salida, junta a otros parámetros, a continuación en la Figura 33 se observa el código de configuración:

```

1  [global_tags]                                #Etiquetas a nivel global
2  project = "TFE"
3
4  [agent]                                       #Configuración del agente
5  interval = "10s"
6  round_interval = true
7  metric_batch_size = 1000
8  metric_buffer_limit = 10000
9  collection_jitter = "0s"
10 flush_interval = "10s"
11 flush_jitter = "0s"
12 precision = "0s"
13 hostname = "telegrafoutput"
14 omit_hostname = false
15
16 [[inputs.mqtt_consumer]]                     #Lee MQTT, en telegraf el que lee MQTT se llama consumer
17 servers = ["tcp://emqx:1883"]                #Conectar al servidor
18
19 topics = [
20   "+/contaminante/#", "+/tiempo/#", "+/temperatura", "+/humedad"
21   #Y se genera el topic donde se guardara en este caso se guardara en temperatura (#para cualquier topic que tenga esta cadena)
22 ]
23
24 ## Username and password to connect MQTT server.
25 username = "influx"
26 password = "ix21a"
27
28 qos = 0
29 connection_timeout = "30s"
30 #qos = 2                                     #Calidad de servicio
31 data_format = "value"
32 data_type = "float"
33
34 [[outputs.influxdb_v2]]                       #Todos los datos leídos en el consumer ahora se van escribiendo en influx db
35 urls = ["http://influxdb:8086"]
36 token = "javiermontfe-datostfe-auth-token"   #Token anteriormente designado
37 organization = "javiermontfe"               #Organización anteriormente designada
38 bucket = "datostfe"                          #Bucket anteriormente designado

```

**Figura 33.** Docker-compose código configuración Telegraf

Fuente: Elaboración propia con la Visual Studio Code

Como se puede observar se ha configurado para recibir la información de todos los topics de toda la contaminación, así como los tiempos de exposición, temperatura y humedad para llevar un seguimiento y un histórico de los datos.

Una vez configurado el sistema, se consiguen obtener los valores a la base de datos de InfluxDB, estos valores estarán ordenados cronológicamente como se observa en la Figura 34.

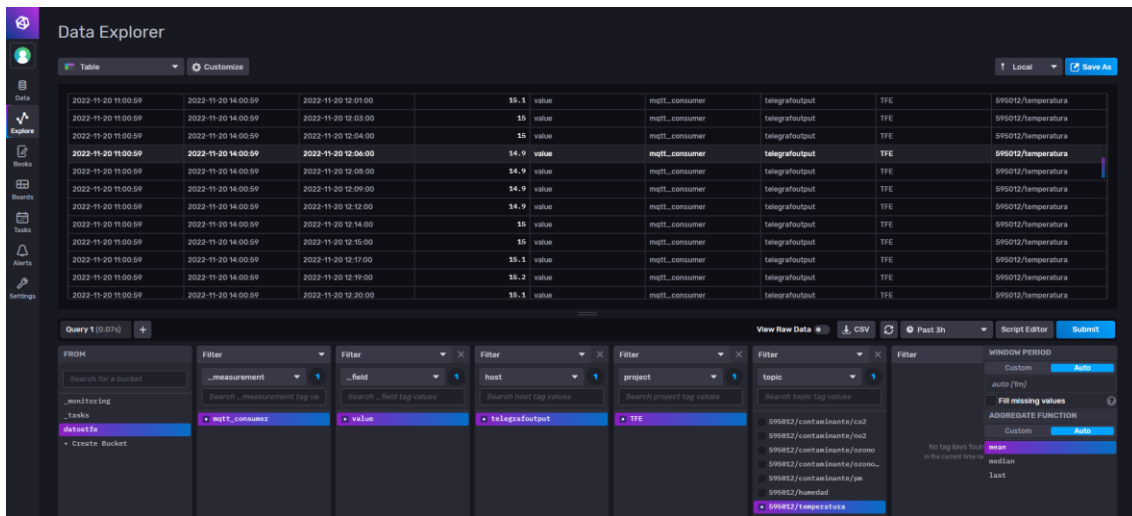


Figura 34. InfluxDB datos de temperatura

Fuente: Elaboración propia con InfluxDB

#### 4.2.3.3. Sistemas de seguridad para clientes

Para garantizar la seguridad se ha configurado en el servidor MQTT usuarios y contraseñas para los diferentes clientes. Para tener un buen nivel de seguridad se han utilizado contraseñas con criptografía Hash SHA256 (Secure Hash Algorithm) con Salt. SHA256 es un algoritmo hash de 256 bits, utilizado para la seguridad criptográfica, al generar hashes irreversibles y únicos, prácticamente imposible de hallar al disponer de una cadena tan grande de combinaciones. También se añade Salt criptográfica que consiste en agregar bits aleatorios al inicio o final de la contraseña para garantizar una contraseña mucho más segura. Para ello se ha generado aleatoriamente una Salt y se le ha añadido a la contraseña del usuario en forma de sufijo. (“Contraseña” + “Salt”) para conjuntamente pasar a encriptación SHA256 y obtener un Hash muy seguro. En la Figura 35 se puede observar el documento csv de usuarios y contraseñas.

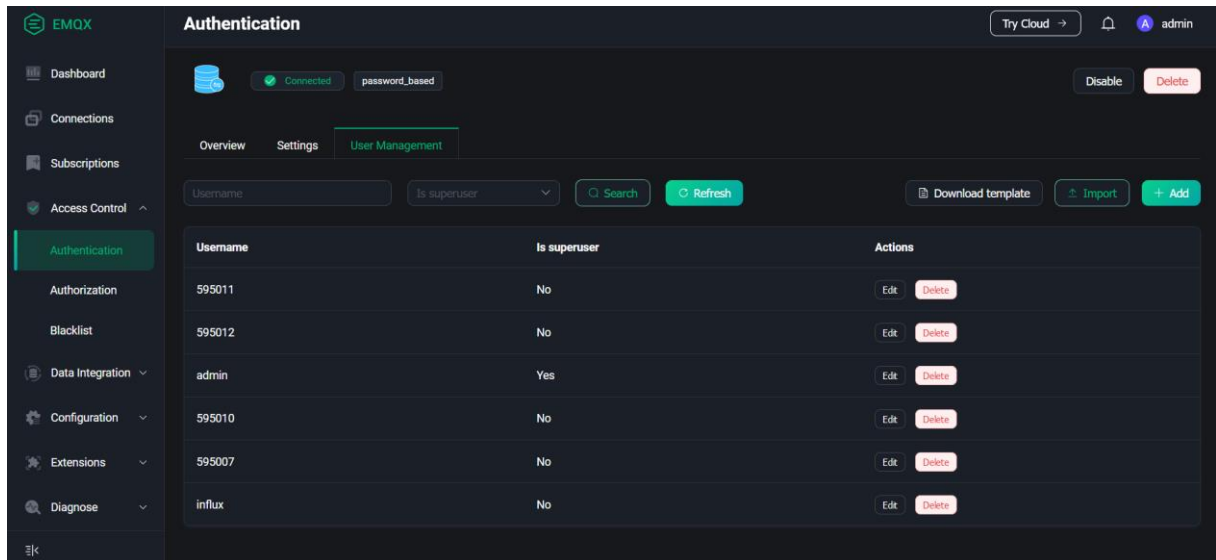
```
user_id,password_hash,salt,is_superuser
admin,731c355274dddc73f0e210fa6bae5cb3fc9a7c8a5d89a90a1ba5db16ccee321,x23fty7o53mcvr0430leornvks13ek59,true
influx,0b36a5bec96b8b807708583fc0a45fa5f41aaa7ad7bb51b191d7ac1e75c51f2e,746p35jkinz78io99dd45kmc032qa23f,false
595012,7baca564b745961f663fb7f45c6c81c865928e5022a8240e270e949d7c78c280,de1024f462fb83910fd13151bd4bd235,false
595011,8ce1e982e6e81f05466e9202e6b9e7e61949947e7004787555e1c1f4f046a056,fa1423f564sbsd84215fd67163g2cf22,false
595010,fad953ed68351a926066ea3d070bec7507242fa3bb507710b568aed5ddded721,dFd3321d434kft5tp3j401f5a4wg88ot,false
595007,d7859c20746436e0d052af1fc0c2077b09150971279f074e9bfc0a0970e0bb,334ghly781pe21xzng6i8pd4w2v77i,false
```

Figura 35. Documento csv para la creación de usuarios y contraseñas seguras

Fuente: Elaboración propia



Para incorporar los usuarios y contraseñas al Broker MQTT se ha importado el archivo csv al Broker. En la Figura 36 se pueden ver los usuarios creados para los diferentes clientes.



**Figura 36.** Usuarios y contraseñas del Broker MQTT

Fuente: Elaboración propia

#### 4.2.3.4. Sistemas de seguridad TLS

Para garantizar la transmisión de datos seguros en la red de comunicación se utiliza una comunicación TLS/SSL para la autenticación unidireccional / bidireccional

Entre las ventajas hay que destacar:

- Certificación sólida: TLS/SSL establece conexiones, con el fin de verificar una entidad de la otra.
- Garantizar la confidencialidad: La comunicación TLS/SSL será encriptada por la clave de la sesión de las dos partes de la conexión.

Para ello ha sido necesario la creación de certificados digitales mediante la herramienta OpenSSL. En este aspecto ha sido necesario crear un certificado CA del servidor junto a su clave.

Para crear el certificado del servidor primero es necesario crear el documento `openssl.cnf` que permite configurar los datos del certificado. Hay que destacar que entre los datos importantes se recalca la IP y el DNS, en la Figura 37 se muestra una captura del código.

```
1 [req]
2   distinguished_name = req_distinguished_name
3   req_extensions = v3_req
4
5   [req_distinguished_name]
6     countryName = Country Name (2 letter code)
7     countryName_default = ES
8     localityName = Locality Name (eg, city)
9     localityName_default = Barcelona
10    organizationalUnitName = Organizational Unit Name (eg, section)
11    organizationalUnitName_default = TFM_javier
12    commonName = Internet Widgits Ltd
13    commonName_max = 64
14
15   [v3_req]
16     basicConstraints = CA:FALSE
17     keyUsage = nonRepudiation, digitalSignature, keyEncipherment
18     subjectAltName = @alt_names
19
20   [alt_names]
21     IP.1 = 127.0.0.1
22     IP.2 = 192.168.1.39
23     DNS.1 = 192.168.1.39
```

**Figura 37.** Configurar `openssl.cnf` para crear solicitud del certificado del servidor MQTT

Fuente: Elaboración propia

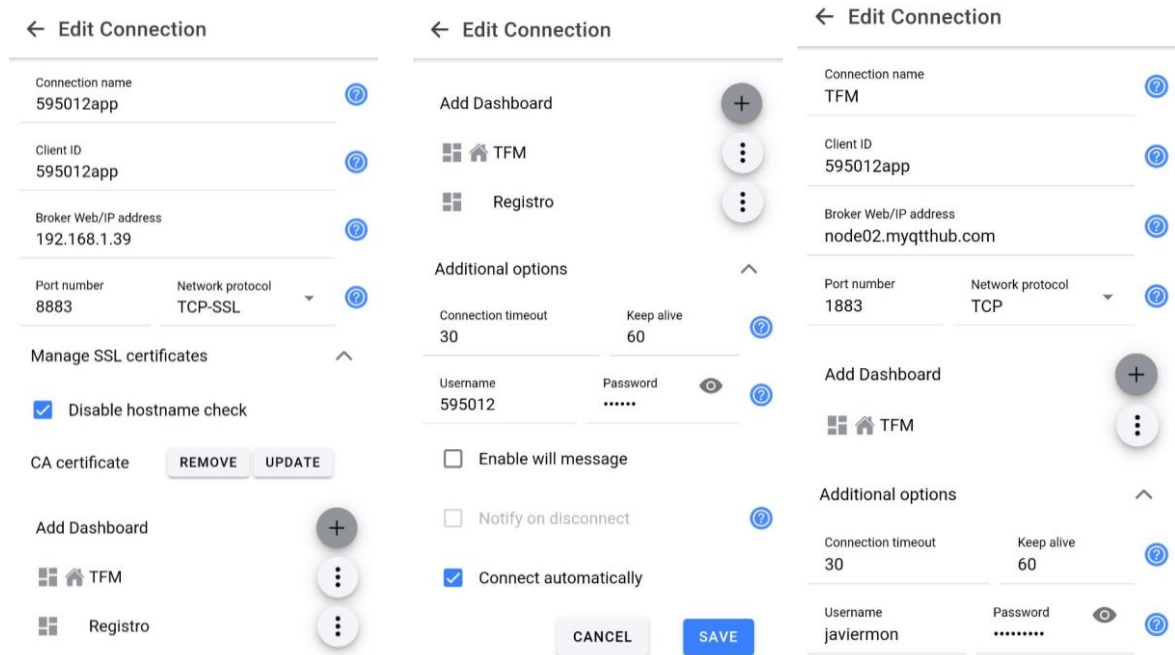
Una vez creado el documento de configuración, a partir de este se crea un archivo `.csr`, la solicitud para el certificado y ya se puede crear el certificado con los datos del CA servidor creado anteriormente, junto a su clave. Y posteriormente se verifica el certificado. También es necesario crear clientes por lo que se genera un CA cliente, creando primero la clave, después generando el archivo `.csr` y el certificado cliente, para ello se volverá a usar el CA certificado del servidor y se verificará.

Finalmente, el cliente también debe tener el certificado a fin de poder hacer una comunicación encriptada. Por lo que se añade un certificado cliente al dispositivo cliente, es decir, al dispositivo de adquisición de datos y a la aplicación para la monitorización.

### 4.3. Diseño y desarrollo del sistema de monitorización y control

#### 4.3.1. Aplicación para el usuario

La aplicación que se utilizará para el desarrollo del Dashboard se trata de la aplicación IoT MQTT Panel disponible para Android, en el Play Store. En la Figura 38 se muestra la configuración de la aplicación con nuestro servidor Broker.



**Figura 38.** Conectar el dispositivo móvil al servidor MQTT

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

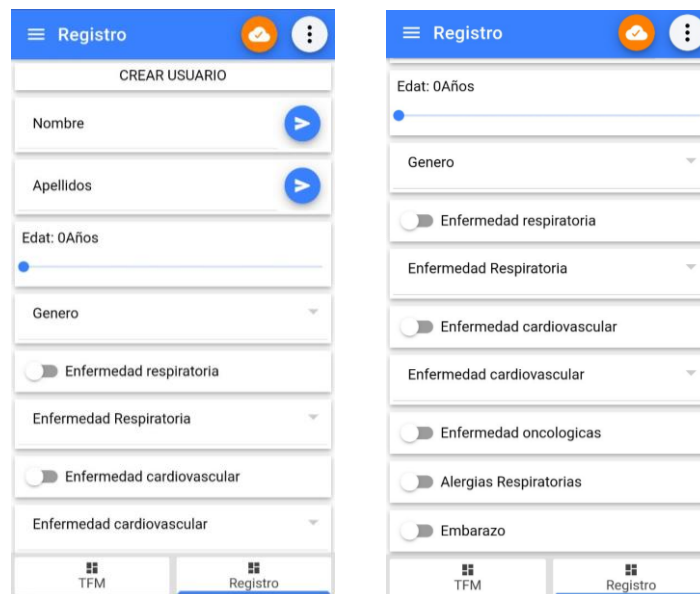
El funcionamiento de la aplicación será el siguiente, en primer lugar, será necesario escribir datos del usuario, en el panel de registro se deberá facilitar los datos personales del usuario, edad, género, enfermedades, alergias o si la persona está embarazada. Esto se seleccionará a través de rellenar los datos en el caso del nombre, un deslizador (slider) para indicar la edad y multi-opciones para el género, ubicación (interior/ exterior) o patologías previas (seleccionar enfermedades), se podrá seleccionar mediante un interruptor.

En el Dashboard de contaminación se tienen unos primeros datos del usuario, las medidas de la temperatura y humedad, las mediciones de los contaminantes en tiempo real y se podrá ver el tiempo de exposición a límites de contaminantes y avisos led en caso de superarlos.

Finalmente, la aplicación también generará avisos en caso de peligro para el usuario. Estos avisos se generan al superar el valor umbral de cada uno de los gases y partículas, marcando que estás en riesgo por ese contaminante. También en algunos casos se indicarán soluciones personalizadas por ejemplo se podrán generar avisos en caso de necesidad de ventilación, entre otros.

En el registro se podrán indicar datos del usuario como son: nombre, apellidos, edad, género, si tiene enfermedades respiratorias, enfermedades cardiovasculares, enfermedades oncológicas, alergias respiratorias o embarazo. Estos datos determinaran si el individuo es de riesgo o no para un contaminante.

En la Figura 39 se muestra una captura de la pantalla de Registro de datos del usuario, para que pueda indicar sus datos:



**Figura 39.** Panel Dashboard para añadir datos del usuario (Nombre, edad, patologías...)

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

En la Figura 40 se muestra una captura del Dashboard monitorización de la contaminación, el tiempo de exposición y los correspondientes avisos leds.

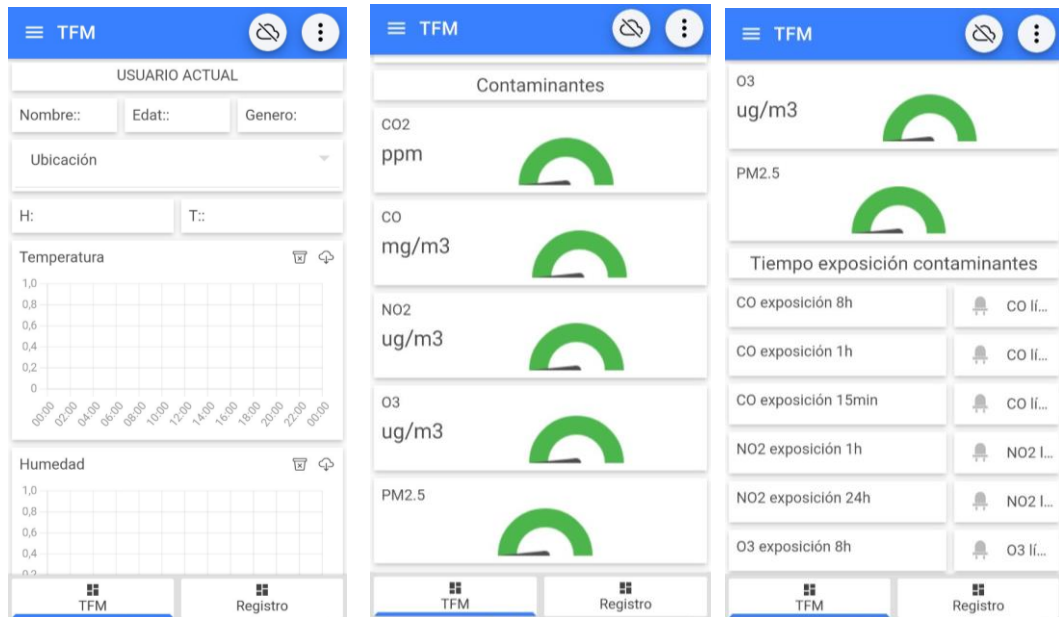


Figura 40. Panel Dashboard para monitorización de la contaminación, tiempo y avisos led

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

Después en de Dashboard de seguimiento se indicará algunos datos del usuario. También se seleccionará la ubicación, mediante 1 o 0 para determinar si se está en interior o exterior, muy importante en el CO<sub>2</sub>. (Figura 41)

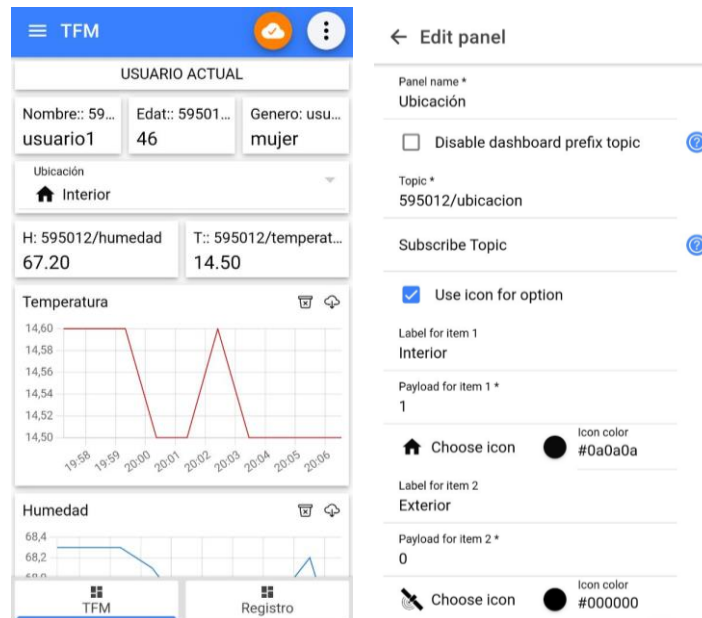
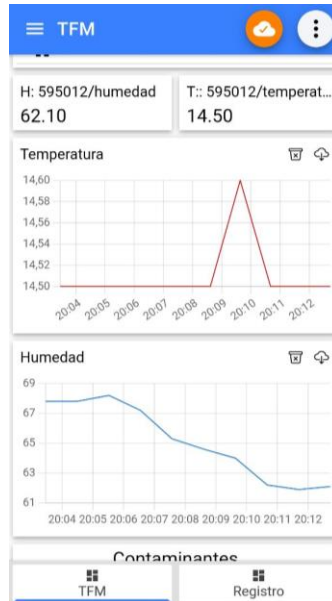


Figura 41. Dashboard datos usuarios y ubicación (izquierda) y configurar panel ubicación (derecha)

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

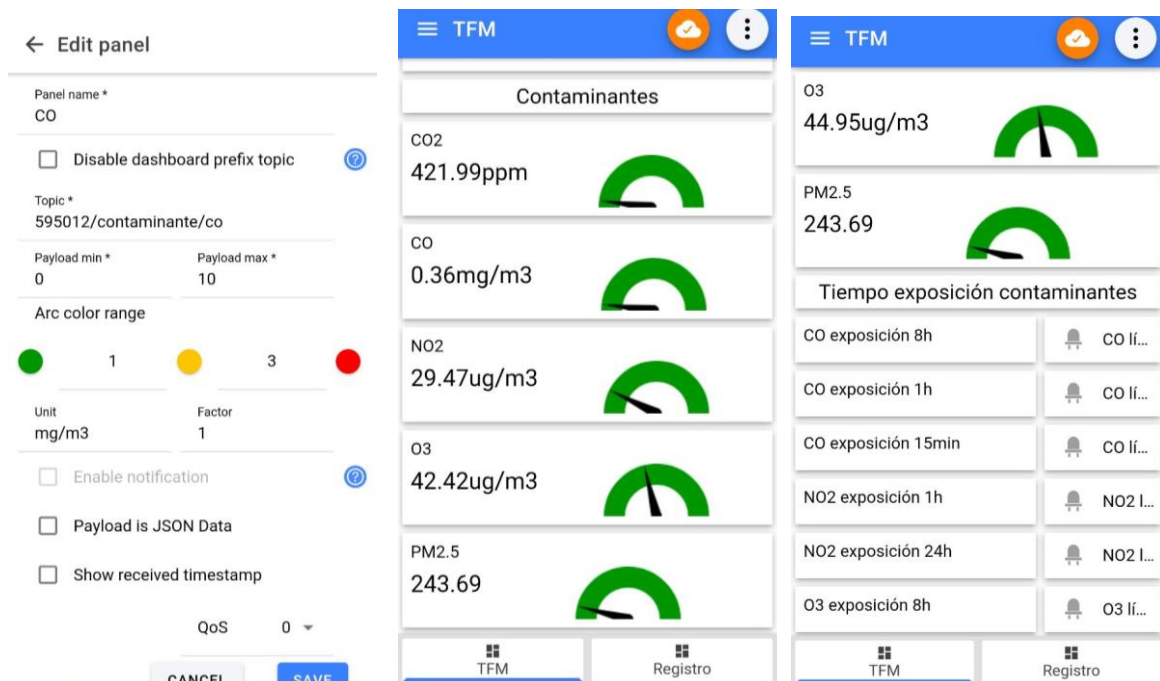
También se logra un seguimiento de la temperatura y humedad como se observa en la Figura 42.



**Figura 42.** Panel Dashboard temperatura y humedad

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

En la Figura 43 se puede observar el diseño y visualización del Dashboard para la monitorización de los gases en tiempo real.

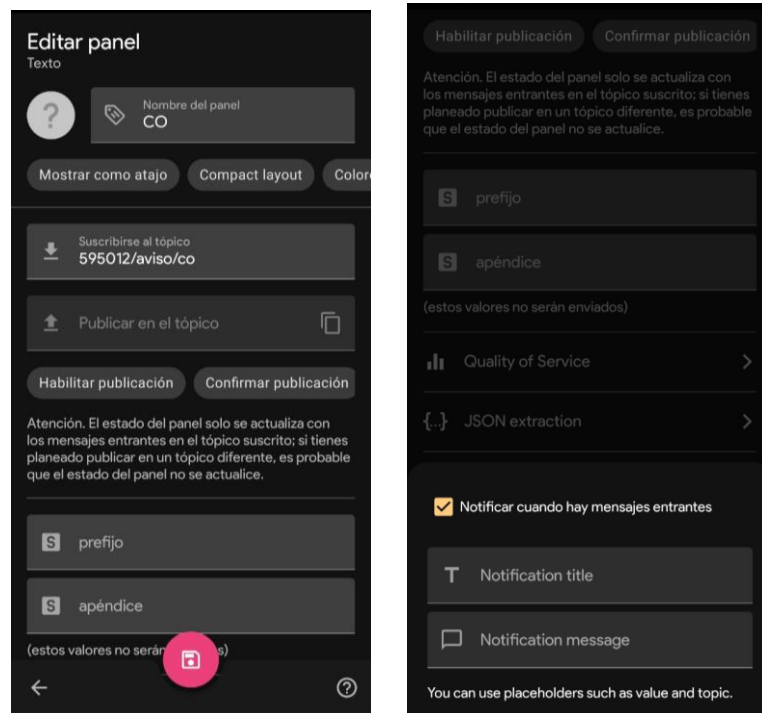


**Figura 43.** Panel Dashboard monitorización de gases

Fuente: Elaboración propia con la aplicación IoT MQTT Panel

### 4.3.2. Aplicación de notificación por medio del teléfono móvil

Para la notificación se utilizará otra aplicación disponible en el Play Store. Se trata de MQTT Dashboard. Esta aplicación permite poder recibir notificaciones en caso de que se perciba peligro de salud por alguno de los gases contaminantes, en la Figura 44 se muestra la activación de notificaciones para un topic específico.



**Figura 44.** Activar notificaciones para un topic específico

Fuente: Elaboración propia con la aplicación MQTT Dashboard

### 4.3.3. Análisis de datos históricos

Es importante para el proyecto tener datos históricos de los contaminantes que han puesto es riesgo al usuario, a continuación, se muestra una captura de InfluxDB con series temporales de CO y así tener los datos históricos del contaminante.

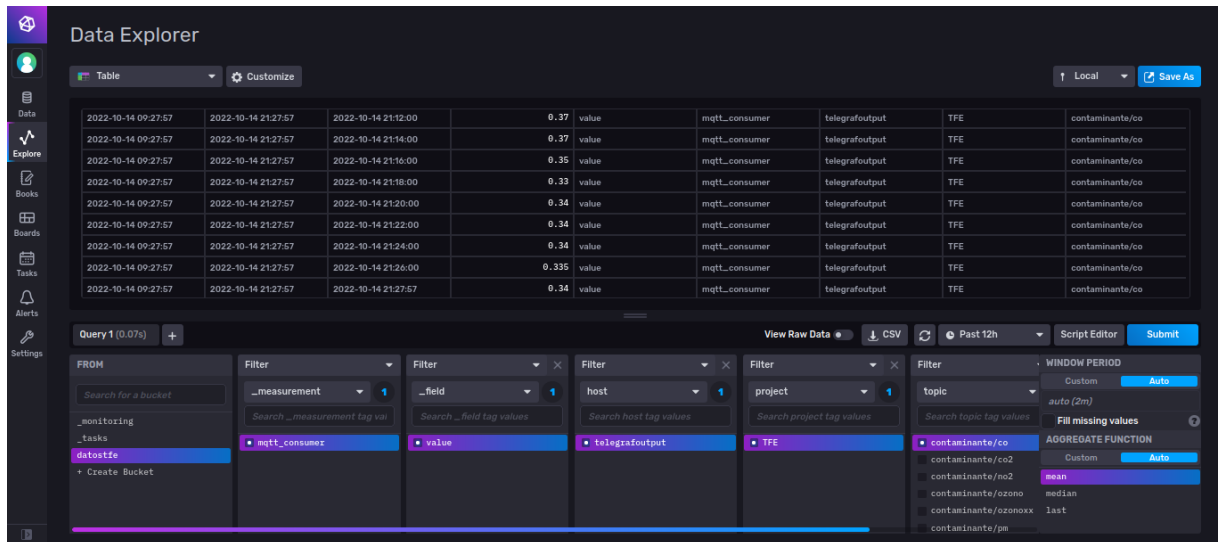


Figura 45. Datos de CO del dispositivo en InfluxDB

Fuente: Elaboración propia con InfluxDB

Se ha configurado en Grafana para que utilice la fuente de datos de InfluxDB asociados al proyecto (Figura 46).

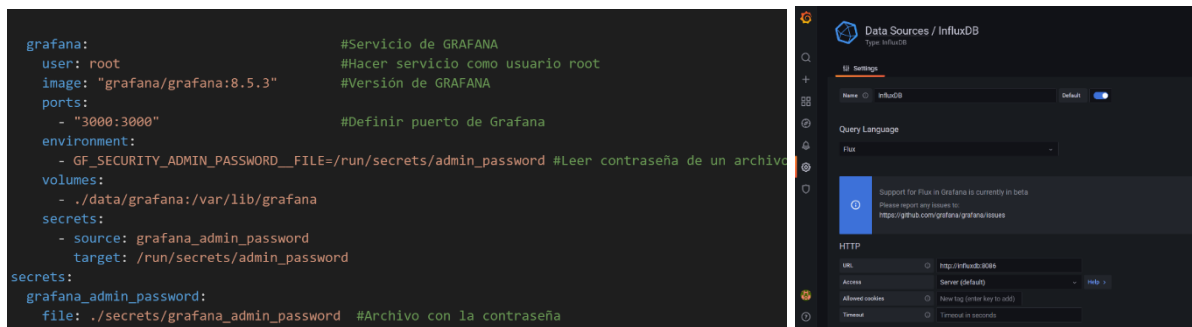
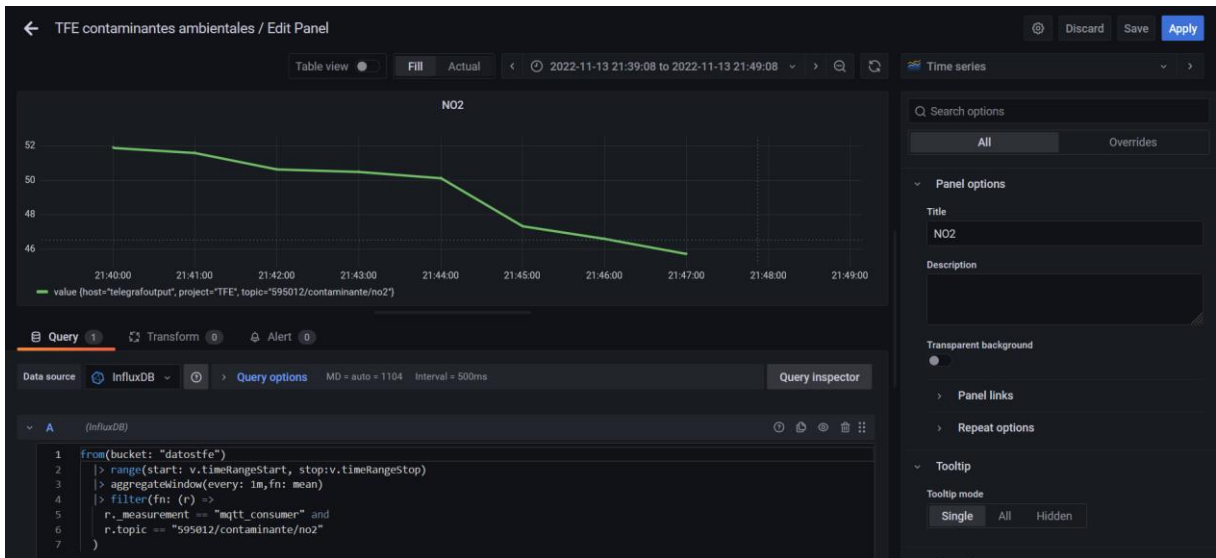


Figura 46. Configurar servicio Grafana (izquierda) y obtener datos de InfluxDB (derecha)

Fuente: Elaboración propia con Grafana

Se ha configurado el Dashboard de Grafana, con los diferentes paneles para una correcta monitorización de la contaminación, formado principalmente por gráficos y gauge. Se han configurado paneles gráficos de datos históricos de contaminación, en la Figura 47 se puede ver un ejemplo de panel gráfico temporal, se ha programado para obtener los datos del bucket y topic específico, en este caso el del contaminante NO<sub>2</sub>, agregando los datos cada minuto:

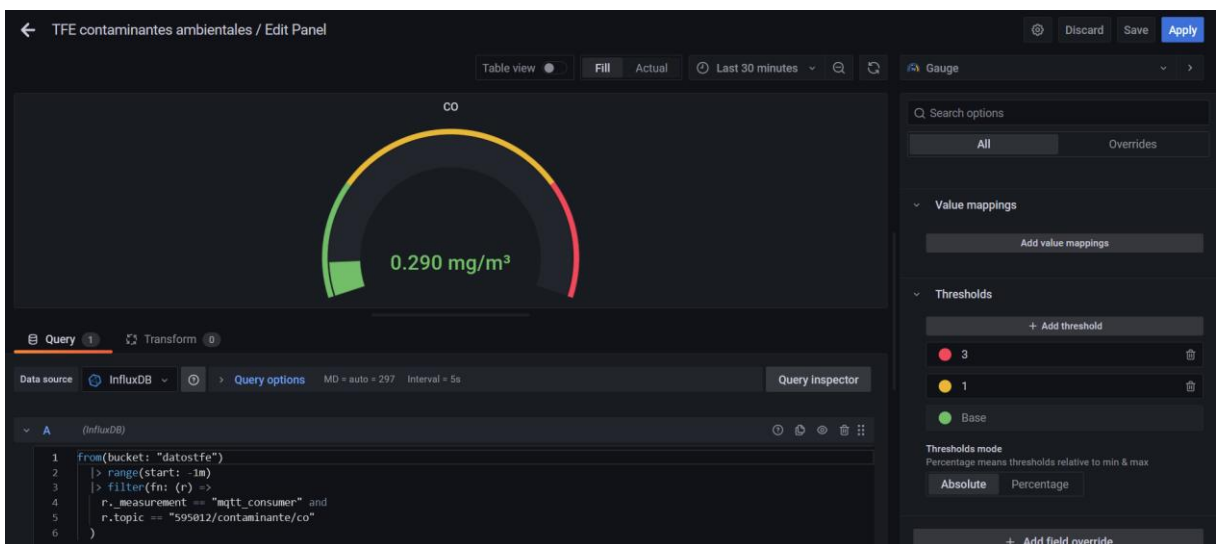




**Figura 47.** Configurar panel de gráfico temporal de Grafana a partir de los datos de InfluxDB

Fuente: Elaboración propia con Grafana

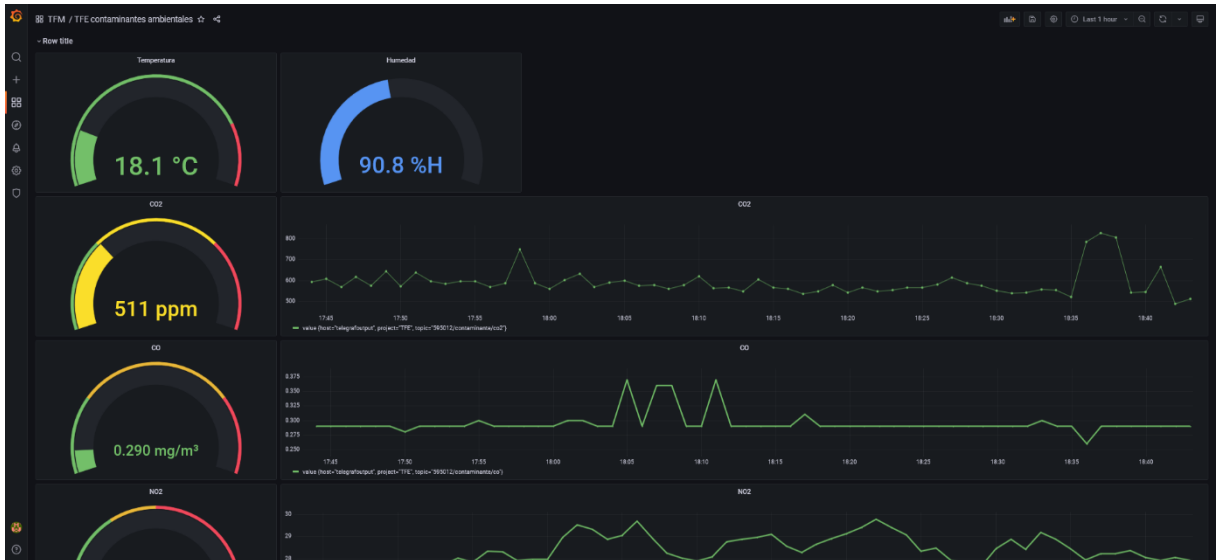
También se han configurado paneles gauge para ver los valores en tiempo real, como se puede observar en la Figura 48 se ha programado para obtener el último dato de InfluxDB del bucket y topic específico, para mostrarlo. También se han podido configurar límites para mostrar el valor en verde si son valores óptimos, amarillo si son valores elevados y rojo si son peligrosos para la salud.



**Figura 48.** Configurar panel de gauge de Grafana a partir de los datos de InfluxDB

Fuente: Elaboración propia con Grafana

Finalmente se muestra en la Figura 49 una captura del Dashboard de Grafana donde se analizan los datos a tiempo real e históricos.



**Figura 49. Dashboard de Grafana**

Fuente: Elaboración propia con Grafana

#### 4.3.4. Creación de una aplicación Android Studio para la monitorización y control

En este subapartado se desarrolla una aplicación para que el usuario/cliente pueda tener en todo momento, en su teléfono móvil personal o tablet, la monitorización y el control de la contaminación atmosférica, de igual manera esta aplicación permite recibir notificaciones debido a la presencia de alta concentración de algún contaminante atmosférico peligroso para la salud del usuario. Se ha creado una aplicación para Android mediante la utilización del Entorno de Desarrollo Integrado (IDE), Android Studio. Para conseguirlo ha sido necesario el desarrollo de todos los elementos de la aplicación Gradle, AndroidManifest, actividades, diseño, maquetación, recursos, entre otros.

Se inicia el desarrollo con la configuración del Gradle, definiendo las características, versión, plugins, personalización de la compilación e implementación de códigos, librerías, entre otros. En la Figura 50 se muestra una parte del código implementado en archivos Gradle

```

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 33
    defaultConfig {
        applicationId "com.example.siscas"
        minSdkVersion 21
        targetSdkVersion 33
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    namespace 'com.example.siscas'
}

```

**Figura 50.** Gradle para el desarrollo de la aplicación

Fuente: Elaboración propia con Android Studio

Seguidamente se configura el AndroidManifest.xml para las características de la aplicación desarrollada, este archivo se encarga de definir la aplicación, permisos, requisitos y al mismo tiempo configura detalles concretos de la aplicación y de cada una de las actividades. En Figura 51 se muestra un fragmento del código del AndroidManifest.

```

<?xml version="1.0" encoding="utf-8">
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat"
        tools:targetApi="31">

        <activity
            android:name=".LoginActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".MainActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">

```

**Figura 51.** AndroidManifest.xml de la aplicación desarrollada

Fuente: Elaboración propia con Android Studio

Para el correcto funcionamiento del protocolo MQTT es necesario que la aplicación funcione como Cliente, para conseguirlo primero se establece la conexión, definiendo el Broker, el ID del Cliente, Usuario, Contraseña y características de la conexión. A continuación, en la Figura 52 se muestra el código para conectar con el Broker:

```

class ClienteMQTT_MainActivity(context: MainActivity) {
    private var mqttClient: MqttAndroidClient

    companion object {
        const val TAG = "AndroidMqttClient"
    }

    fun setCallback(callback: MqttCallbackExtended?) {
        mqttClient.setCallback(callback)
    }

    init {
        mqttClient = MqttAndroidClient(context, MQTT_BROKER, "595012app")
        val options = MqttConnectOptions()
        val sslSocketFactory = SSLSocketFactory.getDefault() //Seguridad SSL
        options.socketFactory = sslSocketFactory
        options.isAutomaticReconnect = true
        options.isCleanSession = true
        options.userName = MQTT_USER
        options.password = MQTT_PASS.toCharArray()
        options.connectionTimeout = 10
        options.keepAliveInterval = 60
        try {
            val caCrtFile = context!!.resources.openRawResource(R.raw.ca)
            options.socketFactory = getSingleSocketFactory(caCrtFile)
            mqttClient.connect(options, null, object : IMqttActionListener {
                override fun onSuccess(asyncActionToken: IMqttToken?) {
                    Log.d(TAG, "Connection success")
                    suscripcionTopics()
                }
                override fun onFailure(asyncActionToken: IMqttToken?, exception: Throwable?) {
                    Log.d(TAG, "Connection failure")
                }
            })
        } catch (e: MqttException) {
            e.printStackTrace()
        }
    }
}

```

**Figura 52.** Código para la conexión MQTT de la aplicación

Fuente: Elaboración propia con Android Studio

Como se ha podido observar en la Figura 52, la comunicación se realiza de forma segura al añadir un certificado CA, permitiendo comunicarse por el puerto seguro 8883, esto es muy importante ya que, en la aplicación se configura la recogida y modificación de datos personales del usuario, siendo necesario que vaya encriptada dado su confidencialidad.

Posteriormente se deben programar las funciones de publicación y suscripción, entre otras funciones útiles y adecuadas para que la aplicación pueda tener una correcta comunicación MQTT con el Broker. A continuación en la Figura 53 se muestra el código de las funciones de publicación y suscripción.

```

fun publish(topic: String, msg: String, qos: Int = 1, retained: Boolean = false) {
    try {
        val message = MqttMessage()
        message.payload = msg.toByteArray()
        message.qos = qos
        message.isRetained = retained
        mqttClient.publish(topic, message, userContext = null, object : IMqttActionListener {
            override fun onSuccess(asyncActionToken: IMqttToken?) {
                Log.d(TAG, "msg published to $topic")
            }

            override fun onFailure(asyncActionToken: IMqttToken?, exception: Throwable?) {
                Log.d(TAG, "Failed to publish $msg to $topic")
            }
        })
    } catch (e: MqttException) {
        e.printStackTrace()
    }
}

private fun suscripcionTopics(){
    try {
        mqttClient.subscribe( topic: "595012/temperatura", qos: 1)
        mqttClient.subscribe( topic: "595012/humedad", qos: 1)
        mqttClient.subscribe( topic: "595012/contaminante/co2", qos: 1)
        mqttClient.subscribe( topic: "595012/contaminante/co", qos: 1)
        mqttClient.subscribe( topic: "595012/contaminante/no2", qos: 1)
        mqttClient.subscribe( topic: "595012/contaminante/ozono", qos: 1)
        mqttClient.subscribe( topic: "595012/contaminante/pm", qos: 1)
        mqttClient.subscribe( topic: "595012/aviso/co2", qos: 1)
        mqttClient.subscribe( topic: "595012/aviso/co", qos: 1)
        mqttClient.subscribe( topic: "595012/aviso/no2", qos: 1)
        mqttClient.subscribe( topic: "595012/aviso/ozono", qos: 1)
    } catch (e: MqttException){
        e.printStackTrace()
    }
}
    
```

Figura 53. Funciones de publicación (izquierda) y suscripción (derecha)

Fuente: Elaboración propia con Android Studio

Otro aspecto para destacar es el diseño de la aplicación, el Layout o maquetación de cada actividad. Para ello se incorporan diferentes elementos, entre los que es necesario destacar el gauge, que permite diferenciar el nivel de contaminación si es peligroso o no, el TextView para ver las publicaciones de topics específicos, el interruptor para activar o desactivar si el usuario se encuentra en interior o exterior, entre otros elementos. A continuación, en la Figura 54 se muestra una parte del código xml junto su visualización.

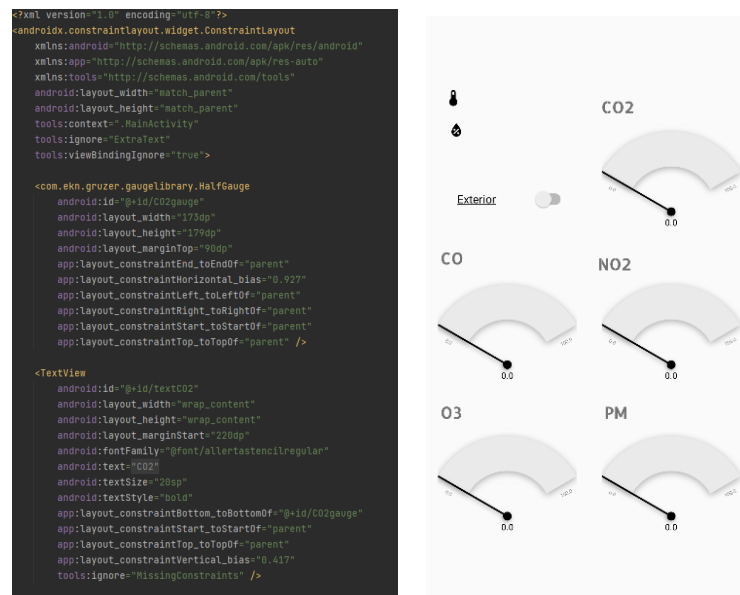


Figura 54. Layout de la medición de contaminación

Fuente: Elaboración propia con Android Studio

A partir del Layout se programa la actividad, programando todas las funcionalidades y ello permite interactuar con los objetos del Layout y añadiendo características como en el caso del gauge permite poner diferentes rangos y colores para que sea más visual, si puede llegar a ser

perjudicial para la salud respiratoria. En la Figura 55 se muestran fragmentos de la programación de la actividad principal, configuración de un elemento gauge, suscripción para que el gauge reciba el valor del dispositivo y publicación de un interruptor.

```

class MainActivity : AppCompatActivity(), NavigationView.OnNavigationItemSelectedListener {
    private val mqttClient by lazy {
        ClienteMQTT_MainActivity(this)
    }

    private lateinit var drawer: DrawerLayout //etiqueta de activity main
    private lateinit var firebaseData: DatabaseReference
    private lateinit var firebaseAuth: FirebaseAuth

    @SuppressWarnings("MissingInflatedId", "WrongViewCast")
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        setMqttCallBack()
        initToolBar()
        initNavigation()

        // CO2 gauge characteristics
        CO2gauge.minValue = 0.0
        CO2gauge.maxValue = 1800.0
        val range_CO2 = Range()
        range_CO2.from = 0.0
        range_CO2.to = 600.0
        range_CO2.color = Color.rgb(90, 255, 57)
        val range2_CO2 = Range()
        range2_CO2.color = Color.rgb(255, 215, 40)
        range2_CO2.from = 600.0
        range2_CO2.to = 1200.0
        val range3_CO2 = Range()
        range3_CO2.color = Color.RED
        range3_CO2.from = 1200.0
        range3_CO2.to = 1800.0
        CO2gauge.addRange(range_CO2)
        CO2gauge.addRange(range2_CO2)
    }

    if(topic == "595012/temperatura"){
        temperatura.text = "Temperatura: $mqttMessage °C"
    }

    if(topic == "595012/humedad"){
        humedad.text = "Humedad: $mqttMessage %H"
    }

    if(topic == "595012/contaminante/co2") {
        CO2gauge.value = "$mqttMessage".toDouble()
        textCO2.text = "CO2: $mqttMessage ppm"
    }

    switch1.setOnClickListener {
        if (switch1.isChecked) try {
            mqttClient.publish("595012/ubicacion", "1")
        } catch (e: MqttException) {
            e.printStackTrace()
        } else {
            try {
                mqttClient.publish("595012/ubicacion", "0")
            } catch (e: MqttException) {
                e.printStackTrace()
            }
        }
    }
}

```

Figura 55. Programación de la actividad principal

Fuente: Elaboración propia con Android Studio

En cada actividad se deben definir las notificaciones, para ello es necesario crear dos funciones una con el canal de envío de la notificación y otra con el mensaje. Este mensaje será el que envíe el dispositivo, esto se consigue extrayendo el Payload cuando se publique un mensaje en el topic "595012/aviso/#". En la Figura 56 se muestra el código para recibir notificaciones y avisos en el teléfono móvil.

```

var CHANNEL_ID= "ALERTA_CONTAMINACION" //Creación del canal de notificación
private fun createNotificationChannel() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) { //Para versiones superiores a la Oreo
        val name = "channel"
        val descriptionText = "channel"
        val importance = NotificationManager.IMPORTANCE_DEFAULT
        val channel = NotificationChannel(CHANNEL_ID, name, importance).apply {
            description = descriptionText
        }
        // Registro del canal como sistema de notificación
        val notificationManager: NotificationManager =
            getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager
        notificationManager.createNotificationChannel(channel)
    }
}

//Creación del mensaje de notificación
private fun notificacion(mensaje:String) {
    var notification = NotificationCompat.Builder(this, CHANNEL_ID)
        .setSmallIcon(R.drawable.chat_alert)
        .setContentTitle("ALERTA")
        .setContentText(mensaje)
        .setPriority(NotificationCompat.PRIORITY_HIGH)
    with(NotificationManagerCompat.from(this)) {
        var notificationId = 1
        notify(/* id = */ notificationId, /* notification = */ notification.build())
    }
}

if (topic == "595012/aviso/co2"){
    var mensaje = "$mqttMessage"
    createNotificationChannel()
    notificacion(mensaje)
}

if (topic == "595012/aviso/co"){
    var mensaje = "$mqttMessage"
    createNotificationChannel()
    notificacion(mensaje)
}

if (topic == "595012/aviso/no2"){
    var mensaje = "$mqttMessage"
    createNotificationChannel()
    notificacion(mensaje)
}

if (topic == "595012/aviso/ozono"){
    var mensaje = "$mqttMessage"
    createNotificationChannel()
    notificacion(mensaje)
}
}

```

Figura 56. Programación de notificaciones

Fuente: Elaboración propia con Android Studio

También se han definido otros Layouts, entre ellos se destaca el de contabilizar el tiempo de exposición a ciertas concentraciones de algunos gases contaminantes, permitiendo visualizar fácilmente si se han superado los límites recomendados por la OMS. También se ha creado Layouts de acceso al usuario, registro y actualizar datos usuarios: nombre, apellidos, edad, si tiene enfermedades respiratorias, cardiovasculares, oncológicas, alergias o embarazo. Estos están conectados a la plataforma de Firebase, permitiendo a la aplicación tener usuario y almacenar los datos de los usuarios. En la Figura 57 se observan capturas de los Layouts, la autenticación y los datos de usuario mediante el uso de la plataforma Firebase.

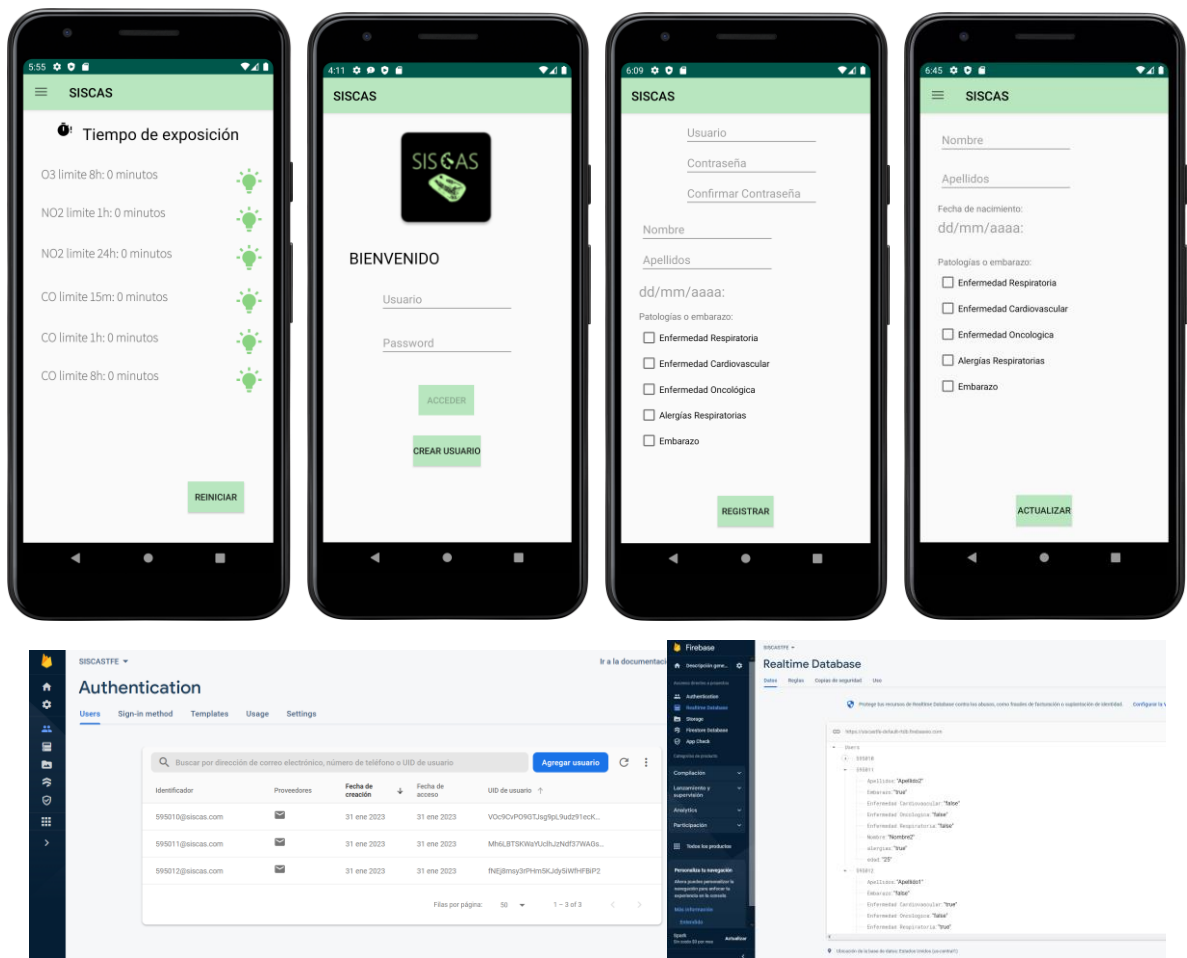
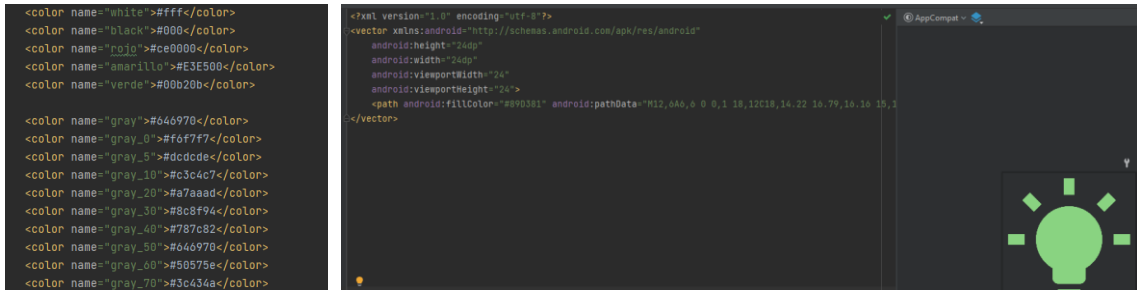


Figura 57. Layouts, autenticación y datos de usuarios en Firebase

Fuente: Elaboración propia con Android Studio y Firebase

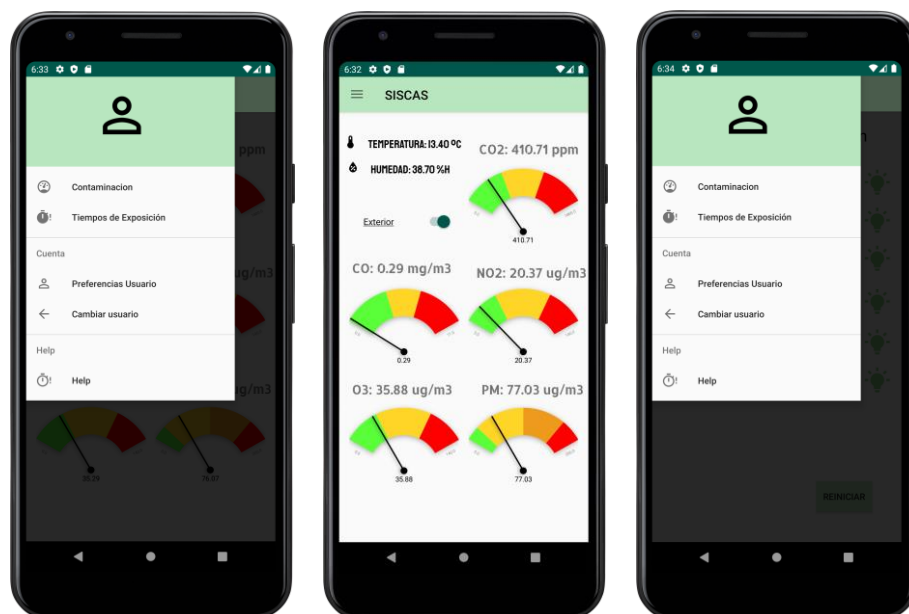
También es importante indicar que mediante la carpeta de recursos se han añadido elementos que hacen más visual la aplicación, mediante iconos, imágenes, colores y Strings. A continuación, en la Figura 58 se muestra parte del código.



**Figura 58.** Muestra para definir colores (izquierda), definir icono de la aplicación (derecha)

Fuente: Elaboración propia con Android Studio

Finalmente, para navegar entre las diferentes actividades se añade un menú, que permite una mejor experiencia al usuario, pudiendo acceder en todo momento a la información deseada. A continuación, en la Figura 59 se muestran capturas del menú de la aplicación y la interface de usuario.



**Figura 59.** Capturas de la aplicación, menú e interface de usuario

Fuente: Elaboración propia con Android Studio

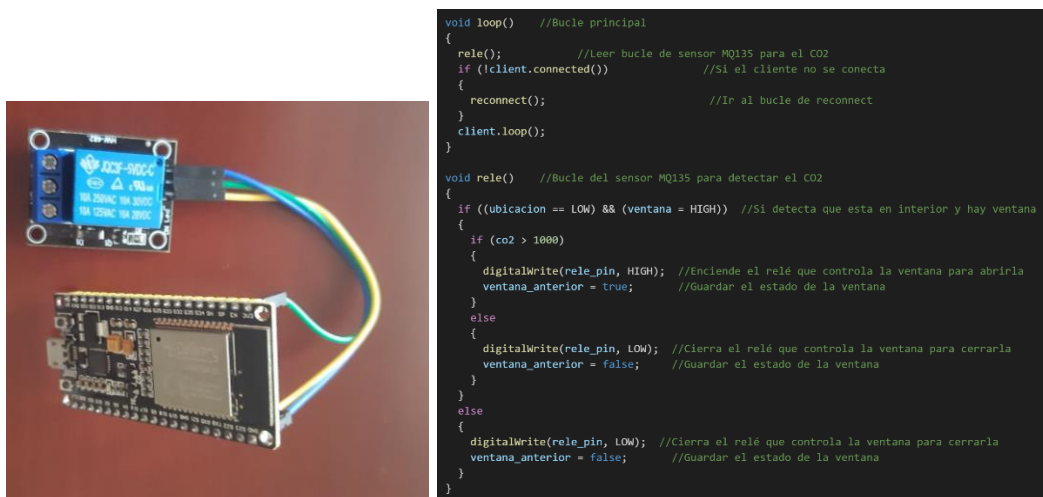
En el Anexo F se puede observar capturas de una parte del código para el desarrollo de la aplicación.



#### 4.3.5. Implementación y solución para interiores

Otra característica que hace diferenciar este proyecto es que también se implementará una funcionalidad dirigida a la concentración de CO<sub>2</sub> en el hogar o interiores, para avisar cuándo sea adecuado aumentar la ventilación del hogar y no sea perjudicial para la salud, así como que automáticamente pueda activar un relé que controle la ventana, para abrirla y reducir el CO<sub>2</sub> dentro de la casa. Esto es muy adecuado para personas mayores, personas con enfermedades neuro-degenerativas (Alzheimer, Parkinson...). También hay que destacar que esta implementación puede ayudar de forma concisa en casos de virus que se transmitan por el aire ya que se ha demostrado que la ventilación ayuda a evitar la transmisión.

La solución de control de ventilación o apertura de ventanas mediante un motor para conseguir limpiar el aire del interior, utilizando un microcontrolador ESP32, un relé y 3 cables. En la Figura 60 se muestra la solución junto al código.



**Figura 60.** Solución para la ventilación en interiores

Fuente: Elaboración propia

El funcionamiento es el siguiente, al detectar que se ha activado ventilación automática y ha indicado que está en interiores, si se detecta una concentración de CO<sub>2</sub> superior al recomendado se activará el relé que controla la ventilación, accionando el dispositivo de apertura de las ventanas. En el anexo B, se muestra la programación del dispositivo de ventilación del hogar.

#### 4.4. Presupuesto y análisis económico

Para la elaboración del prototipo es necesario la compra de distintos componentes. Entre esos componentes es importante destacar sensores, el microcontrolador, así como otros productos como cables, resistencias y una caja.

A continuación, en la Tabla 9 se muestra el presupuesto del dispositivo de adquisición de datos, desglosado y precio total de la construcción del prototipo. Algunos artículos se ha dividido ya que eran packs de un número mayor de unidades de las necesarias.

**Tabla 9.** Presupuesto para la construcción del prototipo

<b>Presupuesto: Prototipo dispositivo de adquisición de datos</b>				
<b>Producto</b>	<b>Descripción</b>	<b>Precio/Uds</b>	<b>Uds</b>	<b>Precio</b>
ESP32	Microcontrolador	10 €/u	1	10 €
DHT22	Sensor de temperatura y humedad	7,99 €/u	1	7,99 €
MQ131	Sensor de Ozono	15 €/u	1	15 €
MQ135	Sensor de calidad de aire CO <sub>2</sub>	4,99 €/u	1	4,99 €
MICS-6814	Sensor de CO,NO <sub>2</sub> y NH <sub>3</sub>	19,99 €/u	1	19,99 €
GP2Y1010AU0F	Waveshare Dust Sensor detector	18,99 €/u	1	18,99 €
Battery Shield	ZDelivery 18650 battery Shield V3	10,99 €/u	1	10,99 €
Bateria 18650	Bateria de litio 18650	8,50 €/u	1	8,50 €
Otros	Cables, Resistencias, caja, Breadboard...	20 €/u	1	20 €
			<b>Resultado</b>	<b>116,45 €</b>

Fuente: Elaboración propia

Modelo de negocio y plan de explotación:

Para poder poner en marcha el producto es necesario tener en mente diferentes costes, para poder ofrecer el producto/servicio. Entre ellos se distingue el coste del dispositivo e implementación. Y otros costes administrativos:

- Dar de alta autónomo 60 euros/mensuales 1 año (reducción del 80% cuota, que en 2022 son 294 euros).
- Modelo TA0521 (añadir DNI + nº afiliación SS)
- Dar de alta en hacienda modelo 036 y concretar IAE – CNAE
- Base de cotización y elegir mutua colaboradora (entra en alta autónomo)
- Oficina española de patentes y marcas, **modelo de utilidad** por un coste 102,39 descuento del 50% emprendedores (baja precio al hacer telemáticamente 87,03 euros)

Durante los primeros años no habrá ningún trabajador asalariado. El modelo de negocio se basará en un único trabajador en régimen de autónomos, siendo este el autor del presente trabajo y el titular del modelo de utilidad. El centro de trabajo estará en el propio domicilio particular y a partir del 2º año se dispondrá un pequeño almacén/trastero de alquiler para material y producto.

La previsión de clientes que adquieran el producto es 150 el primer año y 200 nuevos clientes en el segundo año, todos ellos incluyen la compra del dispositivo de 300 € y el abono mensual de 10€. En la Tabla 10 se muestra la previsión del presupuesto del primer año.

**Tabla 10.** Previsión del presupuesto 1r año

<b>Presupuesto 1º año (150 clientes)</b>	
<b>Estimación 150 clientes</b>	
Venta dispositivo (300 €/u)	45.000 €
Mantenimiento anual (10 €/mes por cliente)	18.000 €
<b>TOTAL INGRESOS</b>	<b>63.000 €</b>
Costes dispositivos	-18.000 €
Coste instalación, montaje y mantenimiento	- 12.000 €
Costes servicios (Servidor, Web y App)	- 3.000 €
Otros costes administrativos	- 103 €
Costes administrativos (autónomos)	- 720 €
Costes modelo utilidad (telematicamente)	- 87 €
<b>TOTAL GASTOS</b>	<b>- 33.910 €</b>
<b>Resultado</b>	<b>29.090 €</b>

Fuente: Elaboración propia

$$ROI = \frac{\text{Facturación} - \text{Inversión}}{\text{Inversión}} \quad (\text{Ec. 2})$$

$$ROI = \frac{63.000 - 33.910}{33.910} = 0,8578$$

Al largo del segundo año se prevé un incremento del coste de fabricación e implementación estimando un IPC de entre 10 al 15%, también hay que tener en cuenta que el precio de las piezas se reduce en proporción al número de dispositivos fabricados, al comprar packs más económicos por la cantidad. De igual manera se incrementan los gastos de Seguridad Social,

almacenaje y otros servicios auxiliares. En la Tabla 11 se muestra la previsión del presupuesto del segundo año.

**Tabla 11.** Previsión del presupuesto 2º año

<b>Presupuesto 2º año</b>	
<b>Estimación 200 clientes (TOTAL 350)</b>	
Venta dispositivo (300 €/u)	60.000 €
Mantenimiento anual (10 €/mes por cliente)	42.000 €
<b>TOTAL INGRESOS</b>	<b>102.000 €</b>
Costes dispositivos	-26.795 €
Coste instalación, montaje y mantenimiento	- 15.000 €
Costes servicios (Servidor, Web y App)	- 1.300 €
Costes administrativos (autónomos)	- 2.200 €
Costes de almacén	- 1.500 €
<b>TOTAL GASTOS</b>	<b>- 46.795 €</b>
<b>Resultado</b>	<b>65.382 €</b>

Fuente: Elaboración propia

$$ROI = \frac{(63.000 + 102.000) - (33.910 + 46.795)}{(33.910 + 46.795)} = 1,0445$$

Al segundo año se prevé la incorporación de 200 clientes, y ya resulta rentable tal y como se observa en el cálculo del ROI al presentar un valor superior al 1.

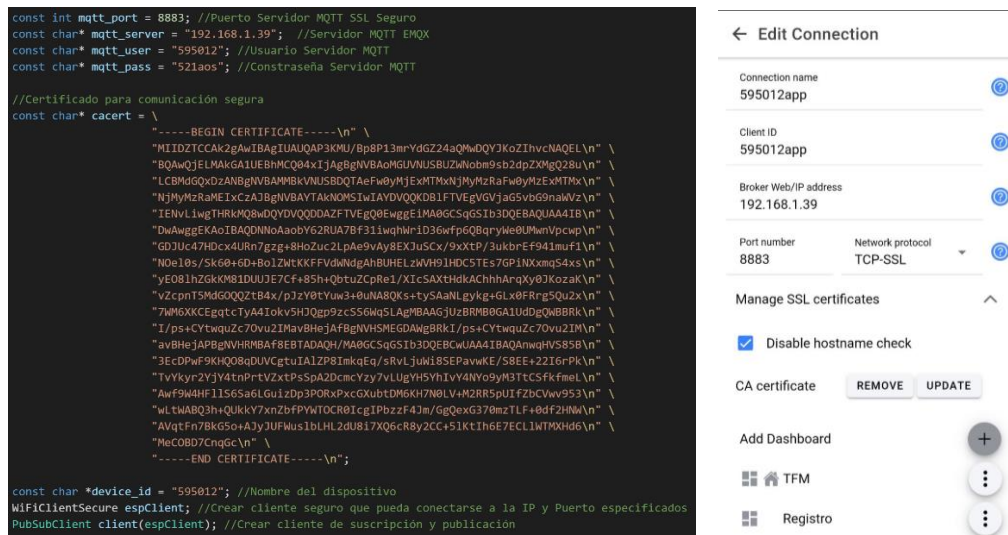
## 5. Evaluación y análisis de resultados

Para la realización de la evaluación se han analizado distintos aspectos del sistema:

- Clientes y su seguridad
- Estructura de control y gestión de los datos
- Evaluación de los datos y del sistema

### 5.1. Clientes y seguridad

Se ha comprobado el acceso de los distintos clientes a fin de asegurar el correcto acceso seguro. A continuación, en la Figura 61 se puede observar el acceso del cliente al Broker con contraseñas y conexión segura SSL/TLS.



**Figura 61.** Configurar usuario y certificados SSL/TLS para el dispositivo (izquierda) y IoT MQTT Panel (derecha)

Fuente: Elaboración propia

También se han realizado pruebas para comprobar la seguridad para esto se ha utilizado Wireshark. En la Figura 62 se puede ver como la información enviada se muestra sin encriptar.

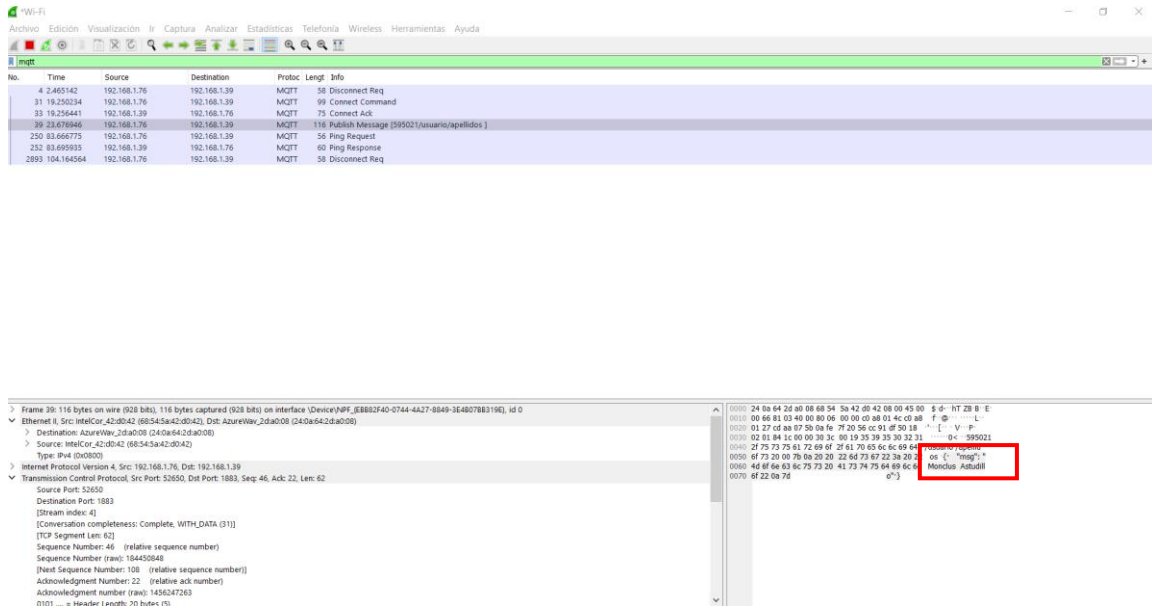


Figura 62. Enviar información MQTT sin SSL/TLS

Fuente: Elaboración propia Wireshark

Como se ha podido observar se puede ver perfectamente la información enviada en el topic “apellidos”. Para solucionar esto mediante el uso de los certificados SSL/TLS creados anteriormente, se puede enviar la información por el puerto 8883 encriptada, en la Figura 63 se puede observar cómo sale encriptado el mensaje. Esto se consigue gracias a los certificados creados.

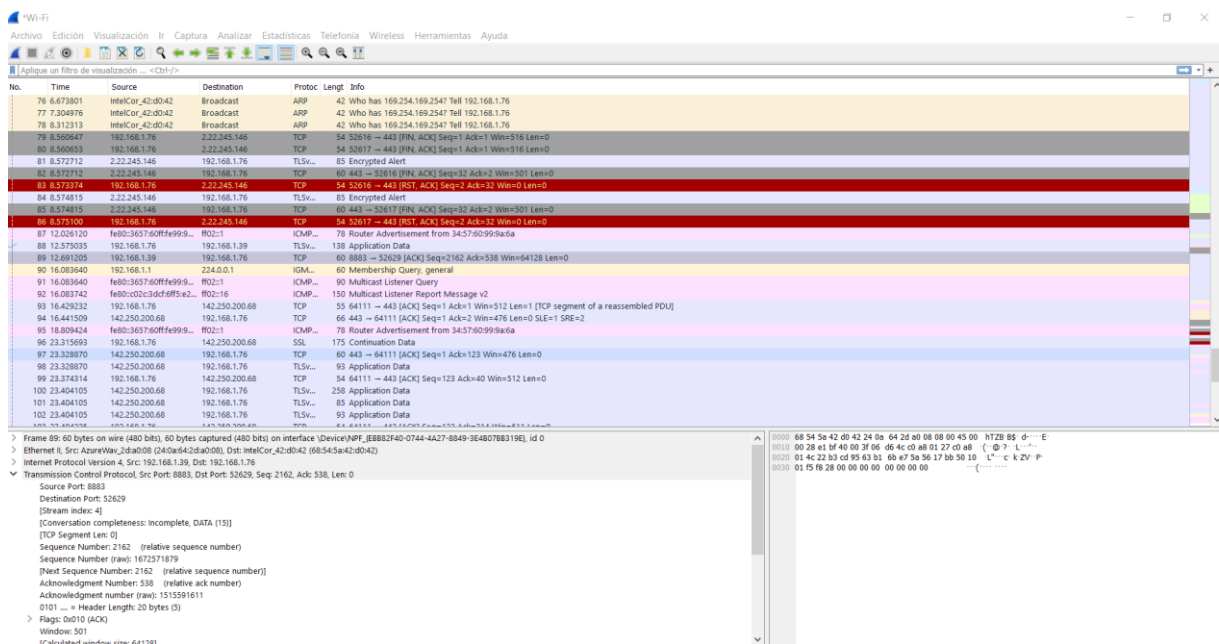


Figura 63. Enviar información MQTT con SSL/TLS (encriptado)

Fuente: Elaboración propia Wireshark

## 5.2. Estructura de control y gestión de los datos

También se ha evaluado la comunicación y Broker MQTT, para ser analizada la gestión de los diferentes clientes, control de acceso, usuarios, contraseñas, topics, entre otros.

En la Figura 64 se puede ver funcionamiento de EMQX y los distintos clientes conectados:

Client ID	Username	Status	IP Address	Keepalive	Connected At	Actions
595012app	595012	Connected	192.168.1.34:48476	60	2022-11-20 15:52:00	Lock Out
Telegram Consumer-4100n	influx	Connected	172.23.0.4:51666	60	2022-11-20 09:25:32	Lock Out
595012n	595012	Connected	192.168.1.34:47306	900	2022-11-20 15:45:08	Lock Out
595012	595012	Connected	192.168.1.88:59091	15	2022-11-20 13:51:18	Lock Out

Figura 64. EMQX clientes conectados

Fuente: Elaboración propia EMQX

También se analiza la comunicación con la base de datos para la gestión de los datos y su almacenamiento. En la Figura 65 se comprueba el almacenamiento de datos en series temporales.

Time	Measurement	Field	Host	Project	Topic	Value
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	571.0699999999999
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	552.87
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	577.885
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	576.22
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	539.37
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	550.1
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	535.215
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	565.205
2022-10-14 09:28:55	mqtt_consumer	value	telegrafoutput	TFE	contaminante/co2	546.54

Figura 65. InfluxDB tabla de datos de CO<sub>2</sub>

Fuente: Elaboración propia InfluxDB

Finalmente InfluxDB permite la creación de gráficos a partir de los datos, como se observa en la Figura 66.

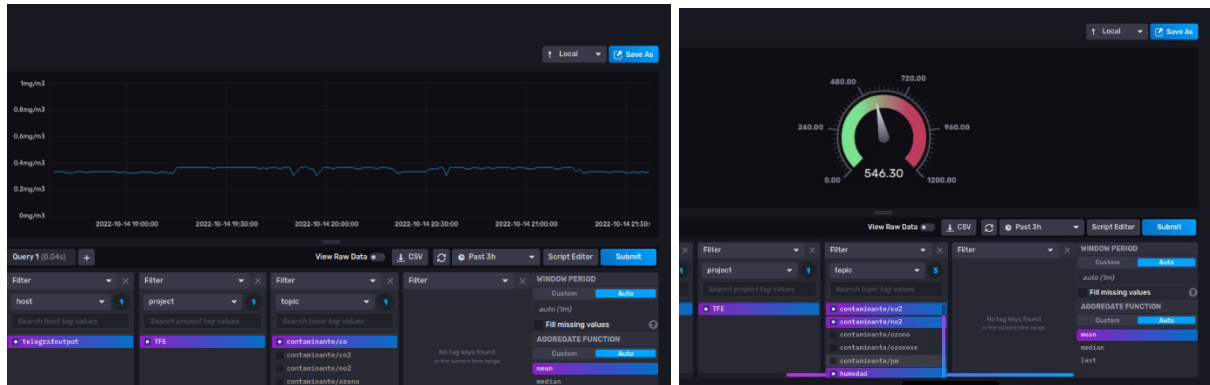


Figura 66. InfluxDB modos representativos de datos

Fuente: Elaboración propia InfluxDB

### 5.3. Evaluación de los datos y del sistema

Para la evaluación de los datos se han comparado los datos obtenidos con los resultados que aportan diferentes webs sobre la contaminación ambiental en distintas ubicaciones en tiempo real. Para comprobarlo se han realizado pruebas cerca de la ubicación de la web oficial de Generalidad de Cataluña para la calidad del aire (Figura 67).

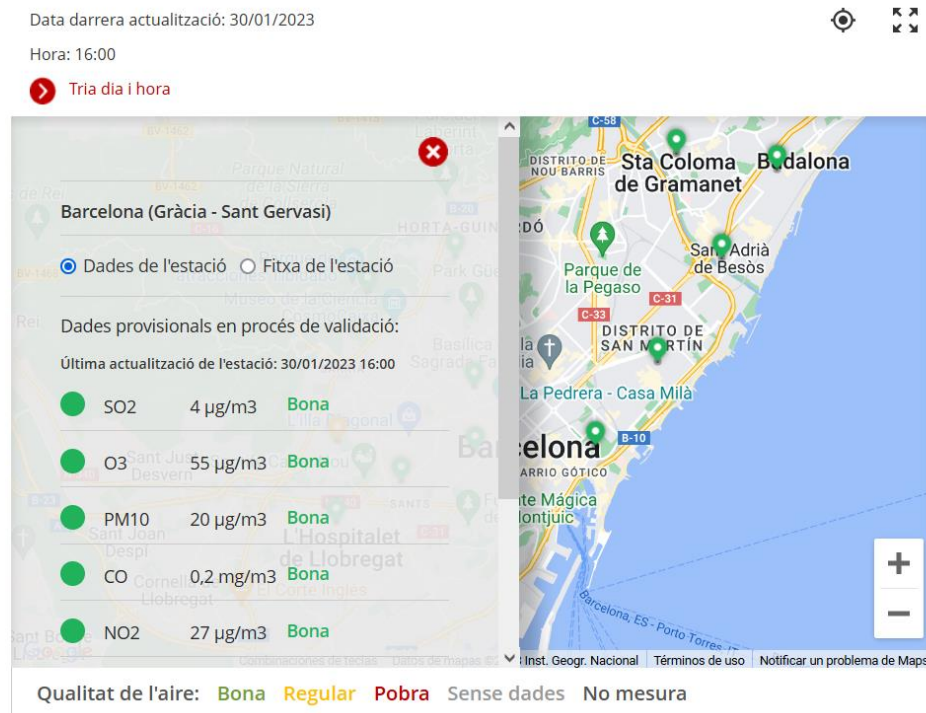
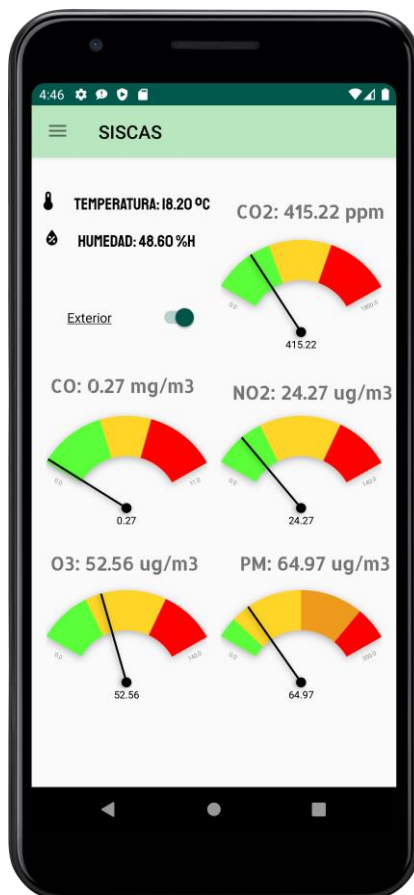


Figura 67. Datos de contaminación en una ubicación cercana a donde se han realizado las pruebas

Fuente: Generalitat de Catalunya, 2022



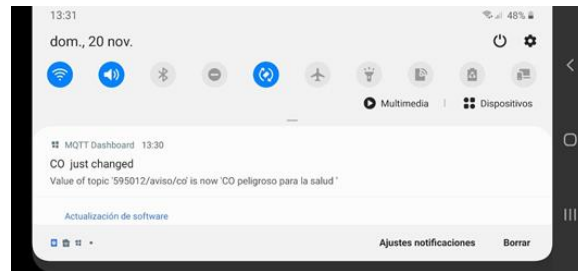
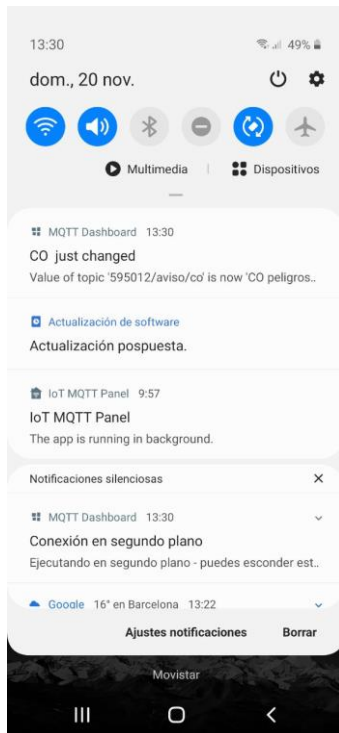
Se ha realizado la comprobación en una ubicación muy cercana al barrio de Barcelona ciudad, en concreto en el barrio de Gracia - Sant Gervasi, como se puede ver en la Figura 68, los resultados son muy parecidos por lo que los sensores actúan de forma correcta, la comunicación con la aplicación y el microcontrolador es correcta y los sensores están bien calibrados.



**Figura 68.** Comprobación valores de algunos de los sensores

Fuente: Elaboración propia con la aplicación propia SISCAS

También se ha comprobado el correcto funcionamiento de los avisos, en este caso se ha bajado el límite de CO para que indique que es peligroso para la salud. En la Figura 69 se puede observar el adecuado funcionamiento las notificaciones. También es importante destacar que la notificación ha sido sonora, pudiendo así alertar al usuario que está ante un nivel de CO peligroso para la salud.



```

if ((e_respiratoria == HIGH) or (edad > 65) or (edad < 15) or (embarazo == HIGH)) //Si la concentración es
{
  if (cof > 0.1) //4
  {
    unsigned long milisegons_co = millis(); //Variable de milisegundos
    if (milisegons_co - previtemps_co > 60000)//Cada minuto
    {
      previtemps_co = milisegons_co; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO peligroso para la salud "); //Publica aviso de CO peligroso
    }
  }
  else {
    if (cof > 0.15) //7
    {
      unsigned long milisegons_co = millis(); //Variable de milisegundos
      if (milisegons_co - previtemps_co > 60000)//Cada minuto
      {
        previtemps_co = milisegons_co; //Actualizar tiempo previo
        client.publish("595012/aviso/co", "CO peligroso para la salud "); //Publica aviso de CO peligroso
      }
    }
  }
}

```

Figura 69. Comprobación de funcionamiento de notificación y modificación del código

Fuente: Elaboración propia con la aplicación (MQTT Dashboard)

Finalmente se ha bajado el valor límite con el fin de ver como se contabilizan los minutos a los que se está expuesto a ciertos gases, para probarlo se ha decidido reducir el límite de CO de 8 horas a mayor de 0.1 y también la publicación de encender led de aviso al superar los 3 minutos, en la Figura 70 se puede observar el cambio en el código.

```

//CONTAR TIEMPO 8 horas
if (cof > 0.1) //Si la concentración es superior 10
{
  unsigned long milisegons_cotiempo = millis(); //Variable de milisegundos
  if (milisegons_cotiempo - previtemps_cotiempo > 60000)//Cuenta cada minuto
  {
    previtemps_cotiempo = milisegons_cotiempo; //Actualizar tiempo previo
    client.publish("aviso/co", "CO muy peligroso para la salud, maximo 8h");
    //cuando se al boton reinicio se actualiza el valor de tiempo exposicion CO a 0
    if (t_co_8h == 0) {
      t_co_8h = t_co_8h + 1;
      preferences.putInt("tiempoco8h", t_co_8h); //Guarda el valor
      char tco8h[8];
      itoa(t_co_8h, tco8h, 10 );
      client.publish("595012/tiempo/co/8h", tco8h); //Publica el valor
    } else {
      t_co_8h = t_co_8h + 1;
      preferences.putInt("tiempoco8h", t_co_8h); //Guarda el valor
      //Serial.printf("State saved: %d \n", t_co_8h); //Imprimir al serial
      char tco8h[8];
      itoa(t_co_8h, tco8h, 10 );
      client.publish("595012/tiempo/co/8h", tco8h); //Publica el valor
      //Incrementar 1 minuto
    }
  }

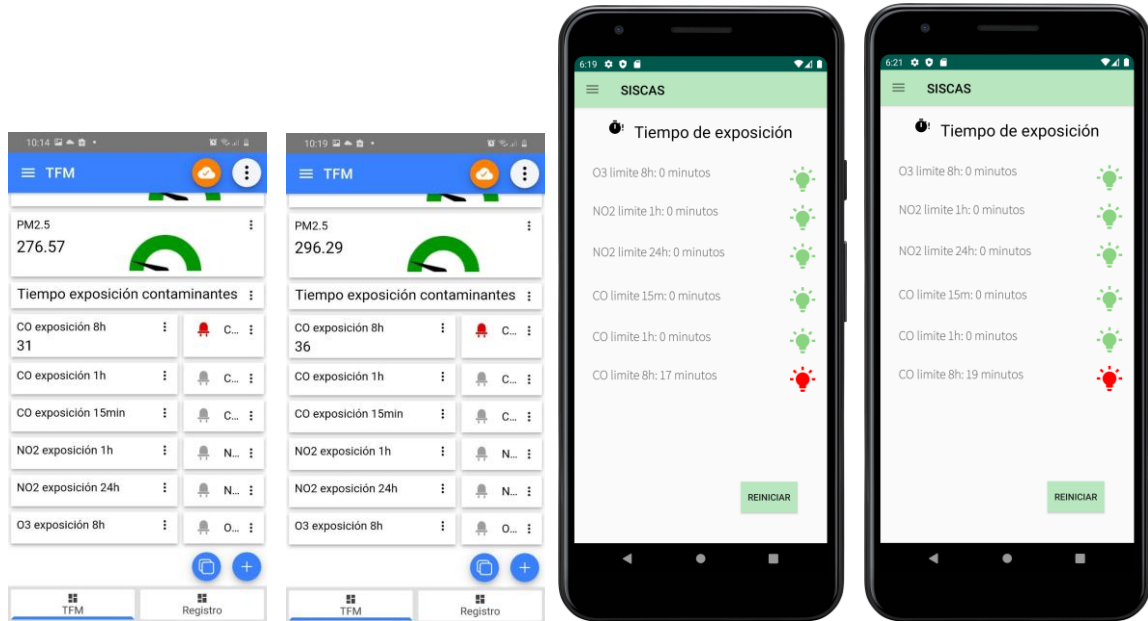
  if (t_co_8h > 3) //Y superior a 8 horas, 480 min
  {
    client.publish("595012/led/co/8h", "1"); //Enciende led de la app para aviso de CO peligroso
  }
}
}

```

Figura 70. Código de funcionamiento del cálculo de tiempo

Fuente: Elaboración propia con Arduino IDE

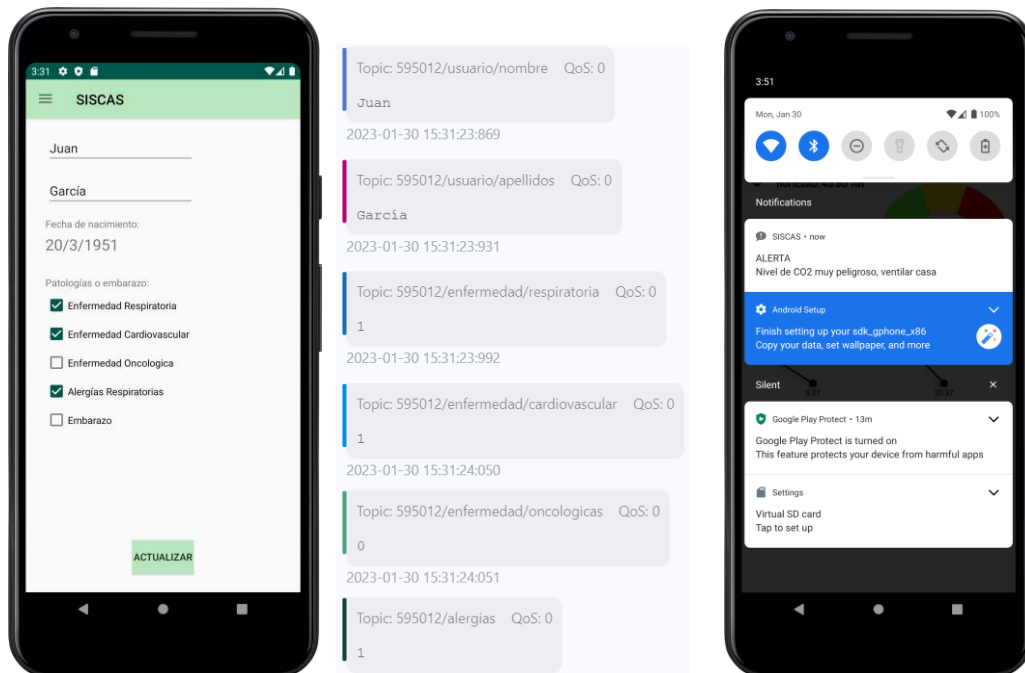
Se han dejado pasar unos minutos, el valor se ha incrementado y se ha cambiado a rojo el led de las aplicaciones (Figura 71):



**Figura 71.** Comprobación de funcionamiento del cálculo de tiempo

Fuente: Elaboración propia con la aplicación IoT MQTT Panel y Android Studio

También se han realizado distintas pruebas para ver el correcto funcionamiento de la aplicación propia SISCAS. En la Figura 72 se puede observar el correcto funcionamiento de registro y envío de datos de usuario y la generación de avisos personalizados.



**Figura 72.** Aplicación SISCAS evaluación registro datos usuario (izquierda) y notificación (derecha)

Fuente: Elaboración propia

Y en la Figura 73 se puede observar un conjunto de capturas de pantalla que demuestran el correcto funcionamiento de la aplicación.

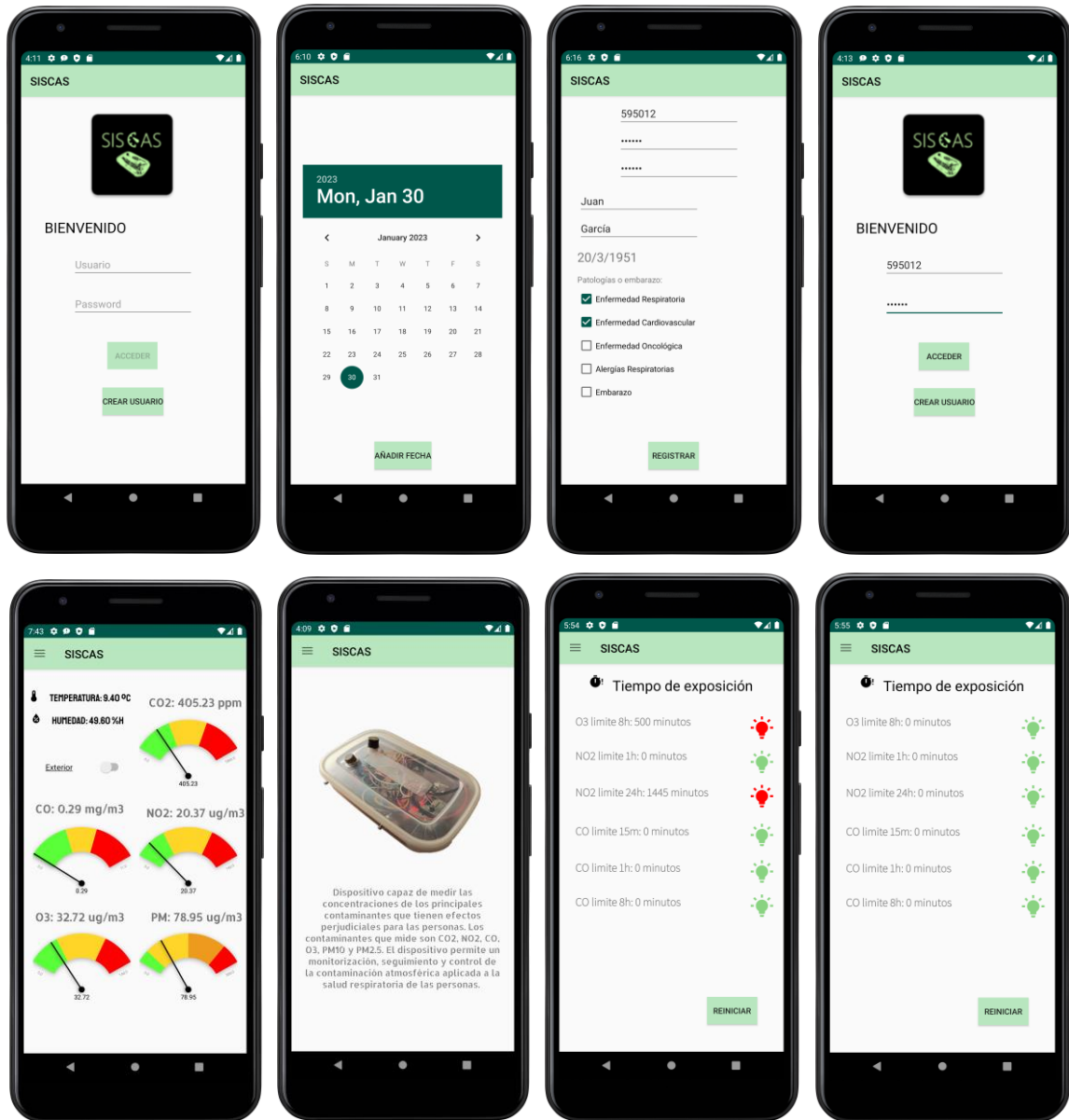
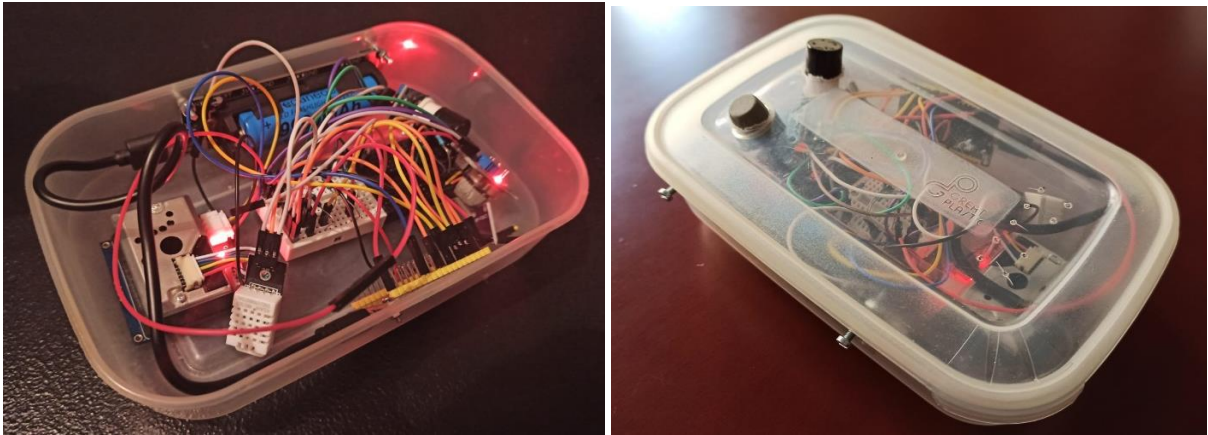


Figura 73. Capturas de la aplicación propia SISCAS

Fuente: Elaboración propia

Finalmente, como se puede observar en la Figura 74 se ha puesto el dispositivo, atornillando los componentes en una caja a fin de que pueda ser transportable y tener un prototipo.



**Figura 74.** Prototipo SISCAS, tapa abierta (izquierda), tapa cerrada (derecha)

Fuente: Elaboración propia

## 6. Conclusiones y trabajos futuros

Tras la elaboración del presente trabajo, una vez observados los resultados obtenidos, se considera que **se ha alcanzado el objetivo general de este proyecto**: Diseñar y desarrollar un sistema de monitorización, seguimiento y control de los gases y partículas de contaminación atmosférica, enfocado en la salud de las personas.

Una vez expuestos los resultados, es posible considerar que, mediante el desarrollo del presente trabajo, **se han alcanzado los objetivos específicos** planteados:

Objetivo específico 1. Diseñar un dispositivo de adquisición de datos de medición de los contaminantes atmosféricos, gases o partículas que afectan a la salud de las personas, tanto en exteriores como interiores. Este objetivo se considera alcanzado tal y como se muestra en el capítulo 4, dando especial importancia al apartado 4.1 con una descripción general de todos los elementos y estructura del dispositivo y el capítulo 5 que muestra la evaluación de los resultados.

Objetivo específico 2. Implementar un sistema de procesamiento, transmisión y tratamiento de datos del dispositivo de adquisición de datos, cumpliendo los requisitos de seguridad. Este objetivo específico se considera alcanzado tal y como muestra en el apartado 4.2 de la memoria, también en la evaluación, en concreto en el apartado 5.2 se observa la correcta comunicación con el Broker y la base de datos.

Objetivo específico 3. Desarrollar un sistema para que el usuario puede añadir información personal (nombre, edad, enfermedades, género, ubicación, entre otros). Este objetivo específico se considera alcanzado tal y como se refleja en el apartado 4.3 de la memoria, en la aplicación se pueden añadir los datos del usuario para que el sistema pueda dar un servicio más personalizado.

Objetivo específico 4. Diseñar un sistema para la visualización y la monitorización de los datos obtenidos. Este objetivo específico se ha cumplido como muestra el apartado 4.3, mediante las aplicaciones y desarrollos que permiten visualizar los datos.

Objetivo específico 5. Generar avisos personalizados cuando los niveles de contaminación son superiores y pueden afectar la salud, teniendo en cuenta los parámetros indicados por la OMS. Este objetivo específico se ha cumplido de manera satisfactoria como se explica en el capítulo

4 de la memoria, tanto en el momento de comunicar los datos del dispositivo en topics específicos, como en la generación de avisos personalizados al móvil, al detectar niveles de contaminación superiores a los recomendados.

El sistema está formado por un dispositivo de adquisición de datos mediante un microcontrolador ESP32 y diferentes sensores capaces de detectar los principales contaminantes atmosféricos que pueden afectar a la salud respiratoria, junto a una fuente de alimentación con el fin que el dispositivo sea móvil y pueda acompañar al usuario en cualquier ubicación donde se encuentre. También incluye un sistema de procesado, transmisión y tratamiento de datos, capaz de gestionar, almacenar y transmitir los datos de forma segura, formado por el Broker EMQX y una base de datos InfluxDB, con un sistema de monitorización y control mediante un Dashboard capaz de mostrar los datos actuales e históricos de la contaminación atmosférica, y al mismo tiempo una aplicación Android que permite al usuario un adecuado seguimiento de la contaminación, registrar sus datos, recibir avisos personalizados, todo en su propio teléfono móvil o tablet.

**Los resultados y evaluación práctica**, en el capítulo 5, demuestran que el sistema diseñado tiene un funcionamiento correcto, comparando los datos obtenidos con los que se ofrecen en webs oficiales que realizan mediciones en distintas ubicaciones fijas. También genera correctamente avisos personalizados y es capaz de determinar el tiempo de exposición a concentraciones elevadas de contaminantes atmosféricos.

En lo referente al nivel de impacto socioeconómico se debe indicar que genera una mejora en la sociedad, debido a que aporta información relevante para el individuo, a fin de tener el control y estar informado en todo momento de los niveles de contaminación atmosférica de los gases y partículas que pueden dañar la salud respiratoria. Asimismo, puede representar una oportunidad de negocio aprovechable que implique un impacto económico, y social, positivo en la zona.

Finalmente, se puede concluir que, mediante la realización del presente trabajo, **se ha desarrollado la solución para un sistema de seguimiento y control de la contaminación atmosférica aplicada a la salud.**

## 6.1. Líneas de trabajo futuras

Una de las líneas de trabajo futuras sería la implementación de nuevos sensores de diferentes gases, ya no tan comunes en las ciudades, pero que en ciertas zonas industriales podrían ser útiles, con el fin de controlar la contaminación en muchas instalaciones industriales o ganaderas. Entre los gases que se podrían implementar presentes en la industria se pueden destacar el Dióxido de Azufre, Amoniacu u otros metales como Antimonio, Cobre, Zinc y Bario (Sb, Cu, Zn y Ba).

También se podrían añadir nuevas funcionalidades a fin de que permitan al usuario una mayor seguridad en su vida cotidiana, como realizar ejercicio o salir a pasear en las horas de menos contaminación, incluso la capacidad de elegir rutas en función de la contaminación atmosférica.

Finalmente remarcar que para la implementación del prototipo se ha usado una placa de pruebas (*breadboard*), permitiendo modificaciones y la posibilidad de realizar cambios, eso a la vez ha hecho que el tamaño sea más grande de lo deseado. Por este motivo se podría implementar una placa PCB con el fin de reducir tamaño y conseguir una mayor facilidad de transporte para el usuario. Actualmente hay una gran variedad de empresas con las que se podría contactar para conseguir el circuito impreso ideal que permitiría reducir el tamaño del dispositivo de adquisición de datos.



## Referencias bibliográficas

- Adafruit. (2021). Adafruit Industries, Unique & fun DIY electronics and kits. Adafruit. Recuperado 22 de diciembre de 2021, de <https://io.adafruit.com/>
- Aeroqual. (s. f.). Portable Air Monitors. Aeroqual. Recuperado 14 de noviembre de 2021, de <https://www.aeroqual.com/products/s-series-portable-air-monitors>
- Agència de Salut Pública de Barcelona Consorci Sanitari de Barcelona Rico, Marc. (2021). Avaluació de la qualitat de l'aire a la ciutat de Barcelona. Informe. 2020. Agència de Salut Pública de Barcelona. <https://bcnroc.ajuntament.barcelona.cat/jspui/handle/11703/122665>
- Ajuntament de Barcelona. (2021). BCNROC. Repositori Obert de Coneixement de l'Ajuntament de Barcelona: Avaluació de la qualitat de l'aire a la ciutat de Barcelona. Recuperado 10 de diciembre de 2021, de <https://bcnroc.ajuntament.barcelona.cat/jspui/handle/11703/98156>
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, 17(4), 2347–2376. <https://doi.org/10.1109/comst.2015.2444095>
- AlDosari, F. (2017). Security and Privacy Challenges in Cyber-Physical Systems. *Journal of Information Security*, 08(04), 285-295. <https://doi.org/10.4236/jis.2017.84019>
- Alldatasheet (2021). MICS-6814 pdf, MICS-6814 Descripción Electrónicos, MICS-6814 Datasheet, MICS-6814 view ::: ALLDATASHEET : Alldatasheet. Recuperado 18 de diciembre de 2021, de <https://pdf1.alldatasheet.es/datasheet-pdf/view/1354644/SGX/MICS-6814.html>
- Alldatasheet (2021). MQ131 pdf, MQ131 Descripción Electrónicos, MQ131 Datasheet, MQ131 view ::: ALLDATASHEET ::: Recuperado 18 de diciembre de 2021, de <https://pdf1.alldatasheet.es/datasheet-pdf/view/1307644/WINSEN/MQ131.html>
- Alldatasheet (2021). MQ135 pdf, MQ135 Descripción Electrónicos, MQ135 Datasheet, MQ135 view ::: ALLDATASHEET ::: Alldatasheet. Recuperado 16 de diciembre de 2021, de <https://pdf1.alldatasheet.es/datasheet-pdf/view/1307647/WINSEN/MQ135.html>

Alldatasheet (2021). DHT22 pdf, DHT22 Descripción Electrónicos, DHT22 Datasheet, DHT22 view ::: ALLDATASHEET ::: alldatasheet. Recuperado 19 de diciembre de 2021, de <https://pdf1.alldatasheet.es/datasheet-pdf/view/1132459/ETC2/DHT22.html>

Alldatasheet (2021). GP2Y1014AU0F pdf, GP2Y1014AU0F Descripción Electrónicos, GP2Y1014AU0F Datasheet, GP2Y1014AU0F view ::: ALLDATASHEET ::: alldatasheet. Recuperado 18 de diciembre de 2021, de <https://pdf1.alldatasheet.es/datasheet-pdf/view/1243996/SHARP/GP2Y1014AU0F.html>

AMAZON. (2021). Temtop M2000 2nd Monitor de calidad del aire para PM2.5 PM10 Partículas Temperatura CO2 Humedad configurable Alarma de audio Grabación Curva Fácil Calibración (M2000 2end) : Amazon.es: Industria, empresas y ciencia. Recuperado 22 de noviembre de 2021, de [https://www.amazon.es/dp/B07DTNV73T/ref=redir\\_mobile\\_desktop?\\_encoding=UTF8&axitk=f2886636e171af241d655442c0fc4c76&hsa\\_cr\\_id=1285711500502&pd\\_rd\\_plhdr=t&pd\\_rd\\_r=49a888fd-193f-4e8d-87ca-94f033bf0f58&pd\\_rd\\_w=Xuiol&pd\\_rd\\_wg=cYeu1&ref\\_=sbx\\_be\\_s\\_sparkle\\_lsi4d\\_asin\\_0\\_img](https://www.amazon.es/dp/B07DTNV73T/ref=redir_mobile_desktop?_encoding=UTF8&axitk=f2886636e171af241d655442c0fc4c76&hsa_cr_id=1285711500502&pd_rd_plhdr=t&pd_rd_r=49a888fd-193f-4e8d-87ca-94f033bf0f58&pd_rd_w=Xuiol&pd_rd_wg=cYeu1&ref_=sbx_be_s_sparkle_lsi4d_asin_0_img)

ASPL -- Advanced Software Production Line, S.L. (2021). MyQtthub. Recuperado diciembre de 2021, de <https://node02.myqtthub.com/>

Chakroun, M., Sayah, J., Kallab, C. & El-Haddad, S. (2022). Internet of Things: Towards a Solid Ecosystem of Interconnected Things. *Advances in Internet of Things*, 12(03), 35-64. <https://doi.org/10.4236/ait.2022.123004>

Clínic Barcelona. (2021). Cardiopatía Isquémica. Recuperado 22 de diciembre de 2021, de <https://www.clinicbarcelona.org/asistencia/enfermedades/cardiopatia-isquemica>

Clínica Universidad de Navarra. (2021). Asma-bronquial. Recuperado 22 de diciembre de 2021, de <https://www.cun.es/enfermedades-tratamientos/enfermedades/asma-bronquial>

Dust Sensor - Waveshare Wiki. (2022). Waveshare. Recuperado 29 de octubre de 2022, de [https://www.waveshare.com/wiki/Dust\\_Sensor](https://www.waveshare.com/wiki/Dust_Sensor)

Edwards, L., Wilkinson, P., Rutter, G., & Milojevic, A. (2022). Health effects in people relocating between environments of differing ambient air pollution concentrations: A literature

review. Environmental Pollution, 292, 118314.  
<https://doi.org/10.1016/j.envpol.2021.118314>

EEA - European Environment Agency. (2021). Air quality statistics. European Environment Agency. Recuperado 23 de octubre de 2021, de <https://www.eea.europa.eu/data-and-maps/dashboards/air-quality-statistics>

EMQ Technologies Co., Ltd. (2022). Modern Data Infrastructure for IoT. [www.emqx.com](http://www.emqx.com). Recuperado 12 de octubre de 2022, de <https://www.emqx.com/en>

EMQ Technologies Co., Ltd. (2022). Android connects MQTT using Kotlin. [www.emqx.com](http://www.emqx.com). Recuperado 25 de noviembre de 2022, de <https://www.emqx.com/en/blog/android-connects-mqtt-using-kotlin>

EPA - Agencia de Protección Ambiental de Estados Unidos. (2021, 30 marzo). Efectos del material particulado (PM) sobre la salud y el medioambiente. US EPA. Recuperado 23 de noviembre de 2021, de <https://espanol.epa.gov/espanol/efectos-del-material-particulado-pm-sobre-la-salud-y-el-medioambiente>

Espressif. (2021). ESP32-DevKitC Board | Espressif. Recuperado 23 de noviembre de 2021, de <https://www.espressif.com/en/products/devkits/esp32-devkitc>

Funciones de Android Studio, Desarrolladores de Android (2022). Android Developers. Recuperado 23 de noviembre de 2022, de <https://developer.android.com/studio/features?hl=es>

García, A., López, E., Rodríguez, M., Alonso, M., Amaro, S., Arenillas, J.F., Ayo-Martín, O., Castellanos, M., Fuentes, B., Freijo, M.M., Gomis, M., Gómez Choco, M., Martínez P., Morales, A., Palacio, E.J., Segura, T., Serena, J., Vivancos J., Roquer, J. (2021). Recommendations of the Spanish Society of Neurology for the prevention of stroke. Interventions on lifestyle and air pollution. *Neurología (English Edition)*, 36(5), 377-387. doi: 10.1016/j.nrl.2020.05.018

Generalitat de Catalunya. (2022). Vols saber què respires? Gencat - Medi Ambient i Sostenibilitat. Recuperado 10 de febrero de 2022, de [https://mediambient.gencat.cat/ca/05\\_ambits\\_dactuacio/atmosfera/qualitat\\_de\\_laire/vols-saber-que-respires/](https://mediambient.gencat.cat/ca/05_ambits_dactuacio/atmosfera/qualitat_de_laire/vols-saber-que-respires/)

- Grafana Labs. (2022). Grafana: The open observability platform. Recuperado 12 de octubre de 2022, de <https://grafana.com/>
- InfluxData. (2021, 10 diciembre). InfluxDB: Open Source Time Series Database. Recuperado 12 de octubre de 2022, de <https://www.influxdata.com/>
- Instrumentación, I. P. S. L. (s. f.). Medidor de gas AQ-500 | PCE Instruments. Pce-instruments. Recuperado 22 de octubre de 2021, de [https://www.pce-instruments.com/espanol/instrumento-medida/medidor/medidor-de-gases-aeroqual-medidor-de-gas-aq-500-det\\_517035.htm](https://www.pce-instruments.com/espanol/instrumento-medida/medidor/medidor-de-gases-aeroqual-medidor-de-gas-aq-500-det_517035.htm)
- Kaivonen, S., & Ngai, E. C. H. (2020). Real-time air pollution monitoring with sensors on city bus. *Digital Communications and Networks*, 6(1), 23–30. <https://doi.org/10.1016/j.dcan.2019.03.003>
- Khomenko, S., Cirach, M., Pereira-Barboza, E., Mueller, N., Barrera-Gómez, J., Rojas-Rueda, D., de Hoogh, K., Hoek, G., & Nieuwenhuijsen, M. (2021). Premature mortality due to air pollution in European cities: a health impact assessment. *The Lancet Planetary Health*, 5(3), e121-e134. [https://doi.org/10.1016/s2542-5196\(20\)30272-2](https://doi.org/10.1016/s2542-5196(20)30272-2)
- Kwizera, V. D. P. N., Li, Z., Lumorvie, V. E., Nambajemariya, F. & Niu, X. (2021). IoT Based Greenhouse Real-Time Data Acquisition and Visualization through Message Queuing Telemetry Transfer (MQTT) Protocol. *Advances in Internet of Things*, 11(02), 77-93. <https://doi.org/10.4236/ait.2021.112006>
- Leloglu, E. (2017). A Review of Security Concerns in Internet of Things. *Journal of Computer and Communications*, 05(01), 121-136. <https://doi.org/10.4236/jcc.2017.51010>
- Maier, A., Sharp, A., & Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things. *2017 Internet Technologies and Applications (ITA)*. Published. <https://doi.org/10.1109/itecha.2017.8101926>
- Mayo Clinic. (2022a, enero 20). Accidente cerebrovascular - Síntomas y causas. Recuperado 6 de enero de 2022, de <https://www.mayoclinic.org/es-es/diseases-conditions/stroke/symptoms-causes/syc-20350113>

- Mayo Clinic. (2021, 24 septiembre). EPOC - Síntomas y causas. Recuperado 6 de diciembre de 2021, de <https://www.mayoclinic.org/es-es/diseases-conditions/copd/symptoms-causes/syc-20353679>
- Mayo Clinic. (2021a, marzo 23). Cáncer de pulmón - Síntomas y causas. Recuperado 5 de diciembre de 2021, de <https://www.mayoclinic.org/es-es/diseases-conditions/lung-cancer/symptoms-causes/syc-20374620>
- Medlineplus. (2020). Asma. Recuperado 21 de diciembre de 2021, de <https://medlineplus.gov/spanish/ency/article/000141.htm>
- Medlineplus. (2021). Enfermedad de obstrucción pulmonar crónica. Recuperado 1 de diciembre de 2021, de <https://medlineplus.gov/spanish/copd.html>
- Medlineplus. (2021). Neumonía. Recuperado 11 de diciembre de 2021, de <https://medlineplus.gov/spanish/pneumonia.html>
- Ministerio para la Transición Ecológica y el Reto Demográfico. (2021). Evaluación de la Calidad del Aire en España. Informe Anua. Secretaría General Técnica. Centro de Publicaciones. [https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/informeevaluacioncalidadaireespana2020\\_tcm30-529210.pdf](https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/informeevaluacioncalidadaireespana2020_tcm30-529210.pdf)
- Ministerio para la Transición Ecológica y el Reto Demográfico. (2021). Evaluación de la Calidad del Aire en España. Informe Anual 2020. Secretaría General Técnica. Centro de Publicaciones. [https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/informeevaluacioncalidadaireespana2020\\_tcm30-529210.pdf](https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/informeevaluacioncalidadaireespana2020_tcm30-529210.pdf) NIPO: 665-21-045-X
- Ministerio para la Transición Ecológica y el Reto Demográfico. (2021). Efectos en la salud y ecosistemas. MITECO. Recuperado 20 de septiembre de 2021, de <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/salud/>
- Ministerio para la Transición Ecológica y el Reto Demográfico. (2021). Calidad del aire - Metales. MITECO. Recuperado 12 de noviembre de 2021, de <https://www.miteco.gob.es/es/calidad-y-evaluacion-ambiental/temas/atmosfera-y-calidad-del-aire/calidad-del-aire/salud/metales.aspx>

- MyQttHub. (2021). MyQttHub -- Servidor MQTT Cloud -- Alojamiento MQTT. Recuperado diciembre de 2021, de <https://myqttHub.com/>
- Netatmo. (s. f.). Medidor de Calidad del Aire Interior Inteligente. Shop Netatmo. Recuperado 25 de noviembre de 2021, de <https://shop.netatmo.com/es-es/aircare/home-coach/homecoach>
- Ordóñez-Iriarte, J. M. (2020). Salud mental y salud ambiental. Una visión prospectiva. Informe SESPAS 2020. Gaceta Sanitaria, 34, 68–75. <https://doi.org/10.1016/j.gaceta.2020.05.007>
- Organización Mundial de la Salud. (2021, 22 septiembre) Calidad del aire ambiente (exterior) y salud. Recuperado 6 de agosto de 2022, de [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)
- Paterson, C., Sharpe, R., Taylor, T., & Morrissey, K. (2021). Indoor PM2.5, VOCs and asthma outcomes: A systematic review in adults and their home environments. Environmental Research, 202, 111631. <https://doi.org/10.1016/j.envres.2021.111631>
- Patial, S., Nazim, M., Khan, A. A. P., Raizada, P., Singh, P., Hussain, C. M. & Asiri, A. M. (2022, agosto). Sustainable solutions for indoor pollution abatement during COVID phase: A critical study on current technologies & challenges. Journal of Hazardous Materials Advances, 7, 100097. <https://doi.org/10.1016/j.hazadv.2022.100097>
- Peladarinos, N., Cheimaras, V., Piromalis, D., Arvanitis, K. G., Papageorgas, P., Monios, N., Dogas, I., Stojmenovic, M., & Tsaramirsis, G. (2021). Early Warning Systems for COVID-19 Infections Based on Low-Cost Indoor Air-Quality Sensors and LPWANs. Sensors, 21(18), 6183. <https://doi.org/10.3390/s21186183>
- Radouan Ait Mouha, R. A. (2021). Internet of Things (IoT). Journal of Data Analysis and Information Processing, 09(02), 77-101. <https://doi.org/10.4236/jdaip.2021.92006>
- Ratliff, J. (2022, septiembre 9). Home. Docker. Recuperado 12 de octubre de 2022, de <https://www.docker.com/>
- Rivas, E., Santiago, J. L., Martín, F., & Martilli, A. (2022). Impact of natural ventilation on exposure to SARS-CoV 2 in indoor/semi-indoor terraces using CO2 concentrations as a proxy. Journal of Building Engineering, 46, 103725. <https://doi.org/10.1016/j.job.2021.103725>

- Rusu-Zagar, G., Rusu-Zagar, C., Iorga, I., & Iorga, A. (2013). Air Pollution Particles PM10, PM2,5 and the Tropospheric Ozone Effects on Human Health. *Procedia - Social and Behavioral Sciences*, 92, 826–831. <https://doi.org/10.1016/j.sbspro.2013.08.761>
- Sanitas. (2021). Efectos del ozono sobre la salud. Recuperado 23 de septiembre de 2021, de <https://www.sanitas.es/sanitas/seguros/es/particulares/biblioteca-de-salud/prevencion-salud/efectos-ozono.html>
- Schuh, G., Dölle, C., Kantelberg, J., & Menges, A. (2018). Identification of Agile Mechanisms of Action As Basis for Agile Product Development. *Procedia CIRP*, 70, 19–24. <https://doi.org/10.1016/j.procir.2018.02.007>
- Smaya, H. (2022). The Influence of Big Data Analytics in the Industry. *OALib*, 09(02), 1-12. <https://doi.org/10.4236/oalib.1108383>
- Sram, R. J., Binkova, B., Dostal, M., Merkerova-Dostalova, M., Libalova, H., Milcova, A., Rossner, P., Rossnerova, A., Schmuczerova, J., Svecova, V., Topinka, J., & Votavova, H. (2013). Health impact of air pollution to children. *International Journal of Hygiene and Environmental Health*, 216(5), 533–540. <https://doi.org/10.1016/j.ijheh.2012.12.001>
- Sram, R. J. (2020). Impact of Air Pollution on the Health of the Population in Parts of the Czech Republic. *International Journal of Environmental Research and Public Health*, 17(18), 6454. <https://doi.org/10.3390/ijerph17186454>
- Stafoggia, M., Oftedal, B., Chen, J., Rodopoulou, S., Renzi, M., Atkinson, R. W., Bauwelinck, M., Klompmaker, J. O., Mehta, A., Vienneau, D., Andersen, Z. J., Bellander, T., Brandt, J., Cesaroni, G., de Hoogh, K., Fecht, D., Gulliver, J., Hertel, O., Hoffmann, B., . . . Janssen, N. A. H. (2022, enero). Long-term exposure to low ambient air pollution concentrations and mortality among 28 million people: results from seven large European cohorts within the ELAPSE project. *The Lancet Planetary Health*, 6(1), e9-e18. [https://doi.org/10.1016/s2542-5196\(21\)00277-1](https://doi.org/10.1016/s2542-5196(21)00277-1)
- Streule, T., Miserini, N., Bartlomé, O., Klippel, M., & de Soto, B. G. (2016). Implementation of Scrum in the Construction Industry. *Procedia Engineering*, 164, 269–276. <https://doi.org/10.1016/j.proeng.2016.11.619>

- Team Asana. (2021, 27 agosto). Las 12 metodologías más populares para la gestión de proyectos.Asana. Recuperado 22 de septiembre de 2021, de <https://asana.com/es/resources/project-management-methodologies>
- Temtop. (2021). Temtop M2000C 2nd Meter CO2 Sensor with Zero Calibration Function PM2.5 PM10 and Export data. Recuperado 25 de noviembre de 2021, de <https://temtopus.com/products/temtop-m2000c-2nd-generation-air-quality-monitor-pm2-5-pm10-co2-data-export-audio-alarm-easy-calibration>
- 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>
- World Health Organization. (2021). WHO global air quality guidelines: particulate matter (PM2.5 and PM10), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide. World Health Organization. <https://apps.who.int/iris/handle/10665/345329>. Licencia: CC BY-NC-SA 3.0 IGO
- World Health Organization. (2021b, septiembre 22). Air quality database. *Www.Who.Int*. Recuperado 19 de noviembre de 2021, de <https://www.who.int/data/gho/data/themes/air-pollution/who-air-quality-database>
- World Health Organization - Organización Mundial de la Salud. (2021, 22 septiembre). Contaminación del aire doméstico y salud. Organización Mundial de la Salud. Recuperado 1 de octubre de 2021, de <https://www.who.int/es/news-room/fact-sheets/detail/household-air-pollution-and-health>
- World Health Organization. (2021, 22 septiembre). Las nuevas Directrices mundiales de la OMS sobre la calidad del aire tienen como objetivo evitar millones de muertes debidas a la contaminación del aire. WHO. Recuperado 10 de diciembre de 2021, de <https://www.who.int/es/news/item/22-09-2021-new-who-global-air-quality-guidelines-aim-to-save-millions-of-lives-from-air-pollution>
- Yin, H., Zhai, X., Ning, Y., Li, Z., Ma, Z., Wang, X., & Li, A. (2022). Online monitoring of PM2.5 and CO2 in residential buildings under different ventilation modes in Xi'an city. *Building and Environment*, 207, 108453. <https://doi.org/10.1016/j.buildenv.2021.108453>



Yuan, M. (2017, 4 octubre). Conozca MQTT. IBM Developer. Recuperado 12 de noviembre de 2021, de <https://developer.ibm.com/es/articles/iot-mqtt-why-good-for-iot/>

Zu, D., Zhai, K., Qiu, Y., Pei, P., Zhu, X., & Han, D. (2020). The Impacts of Air Pollution on Mental Health: Evidence from the Chinese University Students. *International Journal of Environmental Research and Public Health*, 17(18), 6734. <https://doi.org/10.3390/ijerph17186734>

## Anexo A. Código dispositivo de adquisición de datos

```
#include <Preferences.h> //Librería Preferences para guardar valores
#include <WiFi.h> //Librería de WIFI
#include <DHT.h> //Librería del sensor DHT22
#include <PubSubClient.h> //Librería MQTT (Publicación/Suscripción)
#include <NTPClient.h> //Librería para acceder a la hora actual mediante internet
#include <MQ135.h> //Librería del sensor MQ135
#include <WiFiClientSecure.h> //Librería conexión segura

Preferences preferences; //Iniciar instancia de la librería preferencias
#define DHT_PIN 26 //Definir el pin del sensor DHT22
#define DHT_TYPE DHT22 //Definir modelo sensor DHT22
DHT dht (DHT_PIN, DHT_TYPE); //DHT indicar pin y seleccionar que es el modelo DHT22

const char* ssid = "*****"; //Indicar Red WIFI
const char* password = "*****"; // Contraseña red WIFI
const int mqtt_port = 8883; //Puerto Servidor MQTT SSL Seguro
const char* mqtt_server = "192.168.1.39"; //Servidor MQTT EMQX
const char* mqtt_user = "595012"; //Usuario Servidor MQTT
const char* mqtt_pass = "*****"; //Contraseña Servidor MQTT

//Certificado para comunicación segura
const char* cacert = \
    "-----BEGIN CERTIFICATE-----\n" \
    "MIIDZTCCAK2gAwIBAgIUUAUQAP3KMU/Bp8P13mrYdGZ24aQMwDQYJKoZIhvcNAQEL\n" \
    "BQAwQjEELMAkGA1UEBhMCQ04xIjAgBgNVBAoMGUVNUSBUZWNobm9sb2dpZXMgQ28u\n" \
    "LCBMDGQxDzANBgNVBAMMBkVNUStAeFw0yMjE0MTMxMjE0MTMxMjE0MTMxMjE0\n" \
    "NjMyMzRaMEIxIjAgBgNVBAYTAkNOMSIwIAYDVQQKDBlFTVEgVGVjaG5vbG9naWZ\n" \
    "IENVLiwgTHRkMQ8wDQYDVQQDDAFTVEgQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IB\n" \
    "DwAwggEKAoIBAQNNoAaobY62RUA7Bf31iwqhWriD36wfp6QBqrywe0UMwnVpcw\n" \
    "GDJuc47HDcx4URn7gzg+8HoZuc2LpAe9vAy8EXJusCx/9xxTP/3ukbrE941muf1\n" \
    "N0e10s/Sk60+6D+Bo1ZWtKFFVdWNgAhBUHELzWVH91HDC5TEs7GP1NXxmqS4xs\n" \
    "yE081hZgkM81DUUJE7Cf+85h+QbtuZCpRe1/XIcSAXtHdkAchhhArqXy0JKozaK\n" \
    "vZcpnT5MdGQQZtB4x/pJzY0tYuw3+0uNA8QKs+tySAaNLgykg+GLx0FRrg5Qu2X\n" \
    "7WM6XKCEgqtcTyA4Iokv5HJQgp9zcSS6WqSLagMBAAGjUzBRMB0GA1UdDgQWB\n" \
    "I/ps+CYtwquZc70vu2IMavBHejAFBgNVHSMEGDAWgBRKI/ps+CYtwquZc70vu2IM\n" \
    "avBHejAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAQnqHVS85B\n" \
    "3EeDPwF9KHQ08qDUVCgtuIALZP8ImkqEq/sRvLjuWi8SEPavwKE/S8EE+22I6rPk\n" \
    "TvYkyr2Vj4tnPrtVZxtPsSpA2Dcmcyzy7vLUGvH5YhIvV4NYo9yM3TtCSfkfmeL\n" \
    "Awf9W4HF1LS6Sa6LGuizDp3PORxPxcGXubtDM6KH7N0LV+M2RR5pUIfZbCVwv953\n" \
    "wLtwABQ3h+QUkkY7xnZbFPYWTOCR0IcIPbzZf4Jm/GgQexG370mzTLF+0df2HNW\n" \
    "AVqtFn7BkG5o+AjyJUFwus1bLHL2dU8i7XQ6cR8y2CC+51KtIh6E7ECLlWtMXHd6\n" \
    "MeCOBD7CnqGc\n" \
    "-----END CERTIFICATE-----";

const char *device_id = "595012"; //Nombre del dispositivo
WiFiClientSecure espClient; //Crear cliente seguro que pueda conectarse a la IP y Puerto especificados
PubSubClient client(espClient); //Crear cliente de suscripción y publicación

//*****PINS*****
int dust_pin = 32; //Conexión del sensor Waveshare Dust Sensor Detector Module with Sharp GP2Y1010AU0F
int l_dust_pin = 12; //Conexión del led del medidor de Dust
int pin_dht = 26; // Conexión sensor DHT22
int pin_mq135 = 34; //Conexión sensor MQ135
int pin_mq131 = 39; //Conexión sensor MQ131
int pin_co = 36; //Conexión pin CO del sensor MiCS-6814
int pin_no2 = 35; //Conexión pin NO2 del sensor MiCS-6814

//*****MQ131*****
int lectura_ozono = 0;
float ref_R0 = 0;
float O3_ppm = 0;
float O3_ppb = 0;
float O3_mgm3 = 0;
float O3_ugm3 = 0;

//*****DUST*****
int tiempo_muestra = 280;
int tiempo_dust = 40;
int tiempo_descanso = 9680;
float lectura_pm = 0;
float pm_voltaje = 0;
float pm_resultante = 0;

//*****Variables*****
float h, t, co, no2;
float correctedRZero;
```

```

//*****VARIABLES_APP*****
bool ubicacion = 0; //Variable ubicación interior o exterior
bool e_oncologicas = 0; //Variable enfermedad
bool e_cardiovascular = 0; //Variable enfermedad
bool e_respiratoria = 0; //Variable enfermedad
bool embarazo = 0; //Variable embarazo
int edad = 0; //Variable edad del usuario
int alergias = 0; //Variable alergias
bool t_reinicio = 0; //Variable boton reinicio de contadores

//*****TIEMPO_EXPOSICIÓN*****
unsigned int tiempo3;
unsigned int tiempo2dia;
unsigned int tiempo2hora;
unsigned int tiempo8h;
unsigned int tiempo1h;
unsigned int tiempo15m;
unsigned int tiempodust;

//*****Tiempos_MQTT_Publicación*****
long previtemps_mq131 = 0;
long previtemps_mq135 = 0;
long previtemps_6814 = 0;
long previtemps_no2 = 0;
long previtemps_no2tiempo = 0;
long previtemps_co = 0;
long previtemps_cotiempo = 0;
long previtemps_colhtiempo = 0;
long previtemps_col15mtiempo = 0;
long previtemps_dust = 0;
long previtemps_us = 0;
long previtemps_aozono = 0;

//Descargar la hora mediante internet
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 7200);

//*****PROGRAMA*****
void setup()
{
  Serial.begin(115200); //Configuración del puerto serie
  dht.begin(); //Iniciar el sensor DHT
  preferences.begin("my-app", false); //Iniciar espacio de almacenamiento
  setup_wifi(); //Iniciar la conexión WIFI

  //Obtener valores almacenados
  tiempo3 = preferences.getUInt("tiempoo3", 0);
  tiempo2hora = preferences.getUInt("tiempono2hora", 0);
  tiempo2dia = preferences.getUInt("tiempono2dia", 0);
  tiempo8h = preferences.getUInt("tiempoco8h", 0);
  tiempo1h = preferences.getUInt("tiempoco1h", 0);
  tiempo15m = preferences.getUInt("tiempoco15m", 0);
  tiempodust = preferences.getUInt("tiempodust", 0);

  pinMode(pin_no2, INPUT); //Iniciar pin del sensor MICS-6814 para NO2
  pinMode(pin_co, INPUT); //Iniciar pin del sensor MICS-6814 para CO
  pinMode(pin_mq131, INPUT); //Iniciar pin del sensor MICS-6814 para NO2
  pinMode(pin_mq135, INPUT); //Iniciar pin del sensor MICS-6814 para CO

  espClient.setCACert(cacert);
  client.setServer(mqtt_server, mqtt_port); //Iniciar al servidor MQTT
  client.setCallback(callback); // Inicializar el bucle lectura payloads suscritos
  client.subscribe("595012/ubicacion"); //Suscripción al topic ubicación
  client.subscribe("595012/enfermedad/respiratoria"); //Suscripción al topic enfermedad
  client.subscribe("595012/enfermedad/cardiovascular"); //Suscripción al topic enfermedad
  client.subscribe("595012/enfermedad/oncologicas"); //Suscripción al topic enfermedad
  client.subscribe("595012/embarazo"); //Suscripción al topic embarazo
  client.subscribe("595012/edad"); //Suscripción al topic edad
  client.subscribe("595012/alergias"); //Suscripción al topic alergias
  client.subscribe("595012/tiempo/reiniciar"); //Suscripción al topic reiniciar
  timeClient.begin(); //Iniciar el temps NPT
}

```

```

void setup_wifi()
{
  delay(10); //Esperar 0,01 segundos
  WiFi.begin(ssid, password); // Inicializar WIFI
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500); //Esperar 0,5 segundos
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println(ssid); //Imprimir al serial SSID
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); //Imprimir al serial IP
}

//Bucle lectura de los payload de los topics suscritos
void callback(char* topic, byte* payload, unsigned int length)
{
  String topicStr = topic; //Convertir a string para comparar topic
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.println(topicStr);
  char buff_p[length];
  if (topicStr == "595012/enfermedad/respiratoria") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_respiratoria_p = String(buff_p); //Extraer payload como string
    e_respiratoria = e_respiratoria_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/enfermedad/cardiovascular") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_cardiovascular_p = String(buff_p); //Extraer payload como string
    e_cardiovascular = e_cardiovascular_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/enfermedad/oncologicas") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String e_oncologicas_p = String(buff_p); //Extraer payload como string
    e_oncologicas = e_oncologicas_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/alergias") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String alergias_p = String(buff_p); //Extraer payload como string
    alergias = alergias_p.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/edad") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String edad_p = String(buff_p); //Extraer payload como string
    edad = edad_p.toInt(); //Convertir string a entero
  }
}

```

```
if (topicStr == "595012/embarazo") //Si el topic coincide, hacer lectura
{
  for (int i = 0; i < length; i++)
  {
    buff_p[i] = (char)payload[i];
  }
  buff_p[length] = '\0'; //Seleccionar el último payload del topic
  String embarazo_p = String(buff_p); //Extraer payload como string
  embarazo = embarazo_p.toInt(); //Convertir string a entero
}

if (topicStr == "595012/ubicacion") //Si el topic coincide, hacer lectura
{
  for (int i = 0; i < length; i++)
  {
    buff_p[i] = (char)payload[i];
  }
  buff_p[length] = '\0'; //Seleccionar último payload del topic
  String ubicacion_c = String(buff_p); //Extraer payload como string
  ubicacion = ubicacion_c.toInt(); //Convertir string a entero
}

if (topicStr == "595012/tiempo/reiniciar") //Si el topic coincide, hacer lectura
{
  for (int i = 0; i < length; i++)
  {
    buff_p[i] = (char)payload[i];
  }
  buff_p[length] = '\0'; //Seleccionar el último payload del topic
  String t_reinicio_p = String(buff_p); //Extraer payload como string
  t_reinicio = t_reinicio_p.toInt(); //Convertir string a entero
}
}

void reconnect() // Bucle de reconexión
{
  while (!client.connected()) //Mientras el cliente no este conectado
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect(device_id, mqtt_user, mqtt_pass))//Conectar al client MQTT
    {
      Serial.println("connected");
      client.subscribe("595012/ubicacion"); //Suscripcion al topic ubicación
      client.subscribe("595012/enfermedad/respiratoria"); //Suscripcion al topic enfermedad
      client.subscribe("595012/enfermedad/cardiovascular"); //Suscripcion al topic enfermedad
      client.subscribe("595012/enfermedad/oncologicas"); //Suscripcion al topic enfermedad
      client.subscribe("595012/embarazo"); //Suscripcion al topic embarazo
      client.subscribe("595012/edad"); //Suscripcion al topic edad
      client.subscribe("595012/alergias"); //Suscripcion al topic alergias
      client.subscribe("595012/tiempo/reiniciar"); //Suscripcion tiempo exposicion CO para 15 minutos
      delay(100);
    } else {
      Serial.print("fallo, rc=");
      Serial.print(client.state());
      Serial.println(" Intentar-lo en 5 segundos");
      delay(5000); // Esperar 5 segundos antes de volver a intentar
    }
  }
}

void loop() //Bucle principal
{
  mq135_DHT22(); //Leer bucle de sensor MQ135 para el CO2
  mq131(); //Leer bucle de sensor MQ131 para el O3
  mics6814(); //Leer bucle de sensor MICS6814 para el CO y NO2
  GP2Y1010AU0F(); //Leer bucle de sensor GP2Y1010AU0F para particulas PM
  if (!client.connected()) //Si el cliente no se conecta
  {
    reconnect(); //Ir al bucle de reconnect para reconectar
  }
  client.loop();
}
```



```

void mq135_DHT22() //Bucle del sensor MQ135 para detectar el CO2 y DHT22 para T y H
{
  //Medición temperatura y humedad para calibración
  h = dht.readHumidity(); //Lectura DHT22 humedad
  t = dht.readTemperature(); //Lectura DHT22 temperatura
  float calibracion_zero = 165; //Valor de R0 calibrado
  MQ135 mq135_sensor(pin_mq135, calibracion_zero); //Indicar pin y R0 calibrada
  float suma_correctedPPM = 0;
  float suma_correctedZero = 0;
  float suma_resistence = 0;

  for (int i = 0; i < 5; i++) { //Promedio de 5 lecturas de medida para obtener un valor más preciso
    float rzero = mq135_sensor.getRZero(); //Obtener valor de resistencia
    correctedRZero = mq135_sensor.getCorrectedRZero(t, h); //Obtener valor corregido de resistencia
    float resistance = mq135_sensor.getResistance(); //Obtener valor resistencia
    float ppm = mq135_sensor.getPPM(); //Obtener valor ppm CO2
    float correctedPPM = mq135_sensor.getCorrectedPPM(t, h); //Obtener valor corregido de ppm CO2
    delay (500);
    suma_correctedZero = suma_correctedZero + correctedRZero;
    suma_resistence = suma_resistence + resistance;
    suma_correctedPPM = suma_correctedPPM + correctedPPM;
  }

  float correctedZero_promedio = suma_correctedZero / 5; //Promedio de lecturas de R0
  float resistance_promedio = suma_resistence / 5; //Promedio de lecturas de Resistencia
  float PPM_promedio = suma_correctedPPM / 5; //Promedio de lecturas de PPM de CO2

  //Publicacion de medidas cada minuto
  unsigned long milisegons_mq135 = millis(); //Variable milisegundos para contabilizar el tiempo
  if (milisegons_mq135 - previtemps_mq135 > 60000) //Cada 60 segundos hacer las publicaciones
  {
    delay (500);
    previtemps_mq135 = milisegons_mq135; //Actualizar tiempo previo
    char temperaturaDHT [15]; //Crear char para temperatura
    char humitatDHT [15]; //Crear char para humdad
    dtostrf(h, 2, 2, humitatDHT); // Convertir float h y t a char para publicar
    dtostrf(t, 2, 2, temperaturaDHT);
    client.publish("595012/temperatura", temperaturaDHT); //Publica la temperatura
    client.publish("595012/humedad", humitatDHT); //Publica la humedad
    char zero_res [15]; //Crear char de resultado
    dtostrf(correctedZero_promedio, 4, 2, zero_res); //Convertir de float a char
    client.publish("595012/contaminante/co2/Rzero", zero_res); //Publicar R0 del sensor MQ135
    char resistance_res [15]; //Crear char de resultado
    dtostrf(resistence_promedio, 4, 2, resistance_res); //Convertir de float a char
    client.publish("595012/contaminante/co2/resistence", resistance_res); //Publicar Resistencia de MQ135
    char ppm_res [15]; //Crear char de resultado
    dtostrf(PPM_promedio, 4, 2, ppm_res); //Convertir de float a char
    client.publish("595012/contaminante/co2", ppm_res); //Publicar PPM de CO2 en MQ135

    if (ubicacion == HIGH) //En el caso de estar en el exterior
    {
      if (PPM_promedio >= 1500) //Si la concentración de ppm es superior o igual a 1500
      {
        client.publish("595012/aviso/co2", "Nivel de CO2 muy peligroso"); //Publicar aviso
      }
      if ((PPM_promedio > 700) and (PPM_promedio < 1500)) //Si la concentración de ppm superior a 700 hasta 1500
      {
        client.publish("595012/aviso/co2", "Nivel de CO2 peligroso"); //Publicar aviso
      }
    } else { //en caso de estar en interior
      if (PPM_promedio >= 1500) //Si la concentración de ppm es superior o igual a 1500
      {
        client.publish("595012/aviso/co2", "Nivel de CO2 muy peligroso, ventilar casa"); //Publicar aviso ventilación
        client.publish("595012/co2/ventilacion", "1"); //Publicar aviso ventilación
      } else {
        client.publish("595012/co2/ventilacion", "0"); //Publicar aviso ventilación
      }
    }
    if ((PPM_promedio > 1000) and (PPM_promedio < 1500)) //Si la concentración de ppm superior a 1000 hasta 1500
    {
      client.publish("595012/aviso/co2", "Nivel de CO2 peligroso en la casa, ventilar casa"); //Publicar aviso vent
      client.publish("595012/aviso/co2", "Nivel de CO2 muy peligroso, ventilar casa"); //Publicar aviso ventilación
      client.publish("595012/co2/ventilacion", "1"); //Publicar aviso ventilación
    } else {
      client.publish("595012/co2/ventilacion", "0"); //Publicar aviso ventilación
    }
  }
}
}
}

```



```
void mics6814() //Bucle sensor mics6814 para el CO y NO2
{
  //Medida y calibración
  float sumaco = 0;
  float sumano2 = 0;

  for (int i = 0; i < 5; i++) { //Promedio de 5 lecturas de medida para obtener un valor más preciso
    co = analogRead(pin_co); //Lectura analógica CO
    no2 = analogRead(pin_no2); //Lectura analógica NO2
    delay (500);
    sumaco = sumaco + co;
    sumano2 = sumano2 + no2;
  }

  float co_m = sumaco / 5; //Promedio de 5 lecturas de CO
  float no2_m = sumano2 / 5; //Promedio de 5 lecturas de NO2

  float coppm = pow ((co_m*0.00322), -1.179) * 4.385; //Concentración CO a mg/m3
  float no2ppm = ((pow ((no2_m*0.00322), -1.007) / 6.855)); //Concentración NO2 a ug/m3
  float cof = ((coppm * 28.0101) / 24.5);
  float no2f = ((no2ppm * 46.0055 * 1000) / 24.5);

  //Publicar medidas
  unsigned long milisegons_6814 = millis(); //Variable de milisegundos
  if (milisegons_6814 - previtemps_6814 > 60000) //Cada 60s
  {
    previtemps_6814 = milisegons_6814; //Actualizar tiempo previo
    char no2x[8];
    dtostrf(no2f, 4, 2, no2x);
    client.publish("595012/contaminante/no2", no2x); //Publica concentración NO2 en ug/m3
    char cox[8];
    dtostrf(cof, 4, 2, cox);
    client.publish("595012/contaminante/co", cox); //Publica concentración CO en mg/m3
  }

  //NO2 avisos
  if ((e_respiratoria) or (e_cardiovascular) or (e_oncologicas) or (edad > 65) or (edad < 6) or (embarazo == HIGH))
  //Si es persona de riesgo para el NO2
  {
    if (no2f >= 200) //Limite 1h
    {
      if (t_reinicio == 1) {
        tiempono2hora = 0;
        delay (1000);
        char tno2hora[8];
        itoa(tiempono2hora, tno2hora, 10 );
        client.publish("595012/tiempo/no2/hora", tno2hora);
        t_reinicio = 0;
      } else {
        unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
        if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
        {
          previtemps_no2tiempo = milisegons_no2tiempo;
          client.publish("595012/aviso/no2", "NO2 muy peligroso para la salud, maximo 1h");
          //Publica aviso de peligroso para la salud
          if (t_reinicio == 0) {
            tiempono2hora = 0;
          }
          tiempono2hora++; //Incrementar 1 minuto
          preferences.putInt("tiempono2hora", tiempono2hora); //Guardar valor
          Serial.printf("State saved: %d \n", tiempono2hora); //Imprimir al serial
          char tno2[8];
          itoa(tiempono2hora, tno2, 10 );
          client.publish("595012/tiempo/no2/hora", tno2); //Publica el valor de minutos
          client.publish("595012/aviso/no2", "NO2 peligroso para la salud ");
          if (tiempono2hora > 60)
          {
            client.publish("595012/led/no2/hora", "1"); //Enciende led de la app para aviso de NO2 peligroso
          } else {
            client.publish("595012/led/no2/hora", "0"); //Enciende led de la app para aviso de NO2 peligroso
          }
        }
      }
    }
  }
}
```



```

if (no2f >= 50)
{
  if (t_reinicio == 1) {
    tiempo2dia = 0;
    delay (1000);
    char tno2dia[8];
    itoa(tiempo2dia, tno2dia, 10 );
    client.publish("595012/tiempo/no2/dia", tno2dia);
    t_reinicio = 0;
  } else {
    unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
    if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
    {
      previtemps_no2tiempo = milisegons_no2tiempo;
      client.publish("595012/aviso/no2", "NO2 peligroso para la salud, maximo 1 dia"); //Publica aviso de peligroso para la salud
      //cuando se al boton reinicio se actualiza el valor de tiempo exposicion NO2 a 0
      if (t_reinicio == 0) {
        tiempo2dia = 0;
      }
      tiempo2dia++; //Incrementar 1 minuto
      preferences.putInt("tiempo2dia", tiempo2dia); //Guardar valor
      Serial.printf("State saved: %d \n", tiempo2dia); //Imprimir al serial
      char tno2[8];
      itoa(tiempo2dia, tno2, 10 );
      client.publish("595012/tiempo/no2/dia", tno2); //Publica el valor de minutos
      client.publish("595012/aviso/no2", "NO2 peligroso para la salud ");
      if (tiempo2dia > 1440) //60min*24h
      {
        client.publish("595012/led/no2/dia", "1"); //Enciende led de la app para aviso de NO2 peligroso
      }
    }
  }
}
if ((no2f > 25) and (no2f < 50))
{
  unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
  if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
  {
    previtemps_no2tiempo = milisegons_no2tiempo;
    client.publish("595012/aviso/no2", "NO2 peligroso para la salud"); //Publica aviso de peligroso para la salud
  }
}
else {
  if (no2f >= 200)
  {
    if (t_reinicio == 1) {
      tiempo2hora = 0;
      delay (1000);
      char tno2hora[8];
      itoa(tiempo2hora, tno2hora, 10 );
      client.publish("595012/tiempo/no2/hora", tno2hora);
      t_reinicio = 0;
    } else {
      unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
      if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
      {
        previtemps_no2tiempo = milisegons_no2tiempo;
        client.publish("595012/aviso/no2", "NO2 muy peligroso para la salud, superior a 200, máximo 1h"); //Publica aviso de peligroso
        tiempo2hora++; //Incrementar 1 minuto
        preferences.putInt("tiempo2hora", tiempo2hora); //Guardar valor
        Serial.printf("State saved: %d \n", tiempo2hora); //Imprimir al serial
        char tno2[8];
        itoa(tiempo2hora, tno2, 10 );
        client.publish("595012/tiempo/no2/hora", tno2); //Publica el valor de minutos
        client.publish("595012/aviso/no2", "NO2 peligroso para la salud ");
        if (tiempo2hora > 60)
        {
          client.publish("595012/led/no2/hora", "1"); //Enciende led de la app para aviso de NO2 peligroso
        }
      }
    }
  }
}
if (no2f >= 120)
{
  if (t_reinicio == 1) {
    tiempo2dia = 0;
    delay (1000);
    char tno2dia[8];
    itoa(tiempo2dia, tno2dia, 10 );
    client.publish("595012/tiempo/no2/dia", tno2dia);
    t_reinicio = 0;
  } else {
    unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
    if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
    {
      previtemps_no2tiempo = milisegons_no2tiempo;
      client.publish("595012/aviso/no2", "NO2 muy peligroso para la salud, maximo 1h"); //Publica aviso de peligroso para la salud
      tiempo2dia++; //Incrementar 1 minuto
      preferences.putInt("tiempo2dia", tiempo2dia); //Guardar valor
      Serial.printf("State saved: %d \n", tiempo2dia); //Imprimir al serial
      char tno2dia[8];
      itoa(tiempo2dia, tno2dia, 10 );
      client.publish("595012/tiempo/no2/dia", tno2dia); //Publica el valor de minutos
      if (tiempo2dia > 1440)
      {
        client.publish("595012/led/no2/dia", "1"); //Enciende led de la app para aviso de NO2 peligroso
      }
    }
  }
}
}
}

```

```

if ((no2f > 50) and (no2f < 120))
{
  unsigned long milisegons_no2tiempo = millis(); //Variable de milisegundos
  if (milisegons_no2tiempo - previtemps_no2tiempo > 60000)//Cada minuto
  {
    previtemps_no2tiempo = milisegons_no2tiempo;
    client.publish("595012/aviso/no2", "NO2 peligroso para la salud"); //Publica aviso de peligroso para la salud
  }
}

//CO avisos
if (cof > 100) //Aviso peligro O3 enfermedades de nivel 0
{
  if (t_reinicio == 1) {
    tiempoco15m = 0;
    delay (1000);
    char tco15m[8];
    itoa(tiempoco15m, tco15m, 10 );
    client.publish("595012/tiempo/co/1h", tco15m);
    t_reinicio = 0;
  } else {
    unsigned long milisegons_co15mtiempo = millis(); //Variable de milisegundos
    if (milisegons_co15mtiempo - previtemps_co15mtiempo > 60000)//Cada minuto
    {
      previtemps_co15mtiempo = milisegons_co15mtiempo; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO muy peligroso para la salud, maximo 15min");
      tiempoco15m = tiempoco15m + 1; //Incrementar 1 minuto
      preferences.putInt("tiempoco15m", tiempoco15m); //Guardar valor
      Serial.printf("State saved: %d \n", tiempoco15m); //Imprimir al serial
      char tco15m[8];
      itoa(tiempoco15m, tco15m, 10 );
      client.publish("595012/tiempo/co/15m", tco15m); //Publicar valor de minutos
      if (tiempoco15m > 15)
      {
        client.publish("595012/led/co/15m", "1"); //Enciende led de la app para aviso de CO peligroso
      }
    }
  }
}

//CONTAR TIEMPO 1h
if (cof > 35) //Si la concentración es superior a 35
{
  if (t_reinicio == 1) {
    tiempoco1h = 0;
    delay (1000);
    char tco1h[8];
    itoa(tiempoco1h, tco1h, 10 );
    client.publish("595012/tiempo/co/1h", tco1h);
    t_reinicio = 0;
  } else {
    unsigned long milisegons_colht tiempo = millis(); //Variable de milisegundos
    if (milisegons_colht tiempo - previtemps_colht tiempo > 60000)//Cada minuto
    {
      previtemps_colht tiempo = milisegons_colht tiempo; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO muy peligroso para la salud, maximo 1h");
      tiempoco1h = tiempoco1h + 1; //Incrementar 1 minuto
      preferences.putInt("tiempoco1h", tiempoco1h); //Guardar valor
      Serial.printf("State saved: %d \n", tiempoco1h); //Imprimir al serial
      char tco1h[8];
      itoa(tiempoco1h, tco1h, 10 );
      client.publish("595012/tiempo/co/1h", tco1h); //Publicar valor de minutos

      if (tiempoco1h > 60)
      {
        client.publish("595012/led/co/1h", "1"); //Enciende led de la app para aviso de CO peligroso
      }
    }
  }
}

//CONTAR TIEMPO 8 horas
if (cof > 10) //Si la concentración es superior 10
{
  if (t_reinicio == 1) {
    tiempoco8h = 0;
    delay (1000);
    char tco8h[8];
    itoa(tiempoco8h, tco8h, 10 );
    client.publish("595012/tiempo/co/8h", tco8h);
    t_reinicio = 0;
  } else {
    unsigned long milisegons_cot tiempo = millis(); //Variable de milisegundos
    if (milisegons_cot tiempo - previtemps_cot tiempo > 60000)//Cuenta cada minuto
    {
      previtemps_cot tiempo = milisegons_cot tiempo; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO muy peligroso para la salud, maximo 8h");
      tiempoco8h = tiempoco8h + 1;
      preferences.putInt("tiempoco8h", tiempoco8h); //Guarda el valor
      char tco8h[8];
      itoa(tiempoco8h, tco8h, 10 );
      client.publish("595012/tiempo/co/8h", tco8h); //Publica el valor
    }
  }

  if (tiempoco8h > 480) //Y superior a 8 horas, 480 min
  {
    client.publish("595012/led/co/8h", "1"); //Enciende led de la app para aviso de CO peligroso
  }
}
}

```

```

if ((e_respiratoria == HIGH) or (edad > 65) or (edad < 14) or (embarazo == HIGH))
//Personas de riesgo CO
{
  if (cof > 4) //Si la concentración es superior 4
  {
    unsigned long milisegons_co = millis(); //Variable de milisegundos
    if (milisegons_co - previtemps_co > 60000)//Cada minuto
    {
      previtemps_co = milisegons_co; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO peligroso para la salud "); //Publica aviso de CO peligroso
    }
  }
} else {
  if (cof > 7) //Si la concentración es superior 7
  {
    unsigned long milisegons_co = millis(); //Variable de milisegundos
    if (milisegons_co - previtemps_co > 60000)//Cada minuto
    {
      previtemps_co = milisegons_co; //Actualizar tiempo previo
      client.publish("595012/aviso/co", "CO peligroso para la salud "); //Publica aviso de CO peligroso
    }
  }
}
}

void GP2Y1010AU0F() //Bucle para detectar las particulas
{
  int suma_pm = 0;
  for (int i = 0; i < 5; i++) { //Realizar 10 lecturas cada segundo a fin de obtener un valor más preciso
    digitalWrite(l_dust_pin, LOW); //Activar led del sensor dust
    delayMicroseconds(tiempo_muestra); //Esperar tiempo de muestra
    lectura_pm = analogRead(dust_pin); //Lectura analogica del pin
    delayMicroseconds(tiempo_dust);
    digitalWrite(l_dust_pin, HIGH); //Apagar led del sensor dust
    delayMicroseconds(tiempo_descanso);
    delay (500);
    suma_pm = suma_pm + lectura_pm;
  }

  int promedio_pm = suma_pm / 5; //Realizar la media de 5 lecturas
  pm_voltaje = promedio_pm * (3.3 / 4095.0); //Calculo voltaje a partir de la señal analógica
  pm_resultante = 170 * pm_voltaje - 0.1; //Calculo de densidad a partir del voltaje

  unsigned long milisegons_dust = millis(); //Variable de milisegundos
  if (milisegons_dust - previtemps_dust > 150000)//Cada 150 segundos
  {
    delay (900);
    previtemps_dust = milisegons_dust; //Actualizar tiempo previo
    char pm_res[8];
    dtostrf(pm_resultante, 4, 2, pm_res);
    client.publish("595012/contaminante/pm", pm_res); //Publicar valor de densidad particulas pm
  }

  if ((e_respiratoria) or (e_cardiovascular) or (e_oncologicas) or (embarazo) or (alergias) or (edad < 14) or (edad > 65))
  //Personas de riesgo del PM
  {
    if (pm_resultante >= 250) //Peligrosos para personas de riesgo
    {
      client.publish("595012/aviso/pm", "PM muy peligroso para la salud "); //Publicar aviso peligro
    }

    if ((pm_resultante > 75) and (pm_resultante < 250)) //De leve contaminación a muy peligrosa
    {
      client.publish("595012/aviso/pm", "PM peligroso para la salud "); //Publicar aviso peligro
    }
  } else {
    if (pm_resultante >= 250) //Peligrosos para personas sin riesgo
    {
      client.publish("595012/aviso/pm", "PM muy peligroso para la salud "); //Publicar aviso peligro
    }

    if ((pm_resultante > 150) and (pm_resultante < 250)) //De moderada contaminación a muy peligrosa
    {
      client.publish("595012/aviso/pm", "PM peligroso para la salud "); //Publicar aviso peligro
    }
  }
}
}
}

```

## Anexo B. Código dispositivo ventilación en el hogar

```
#include <WiFi.h> //Librería de WIFI
#include <PubSubClient.h> //Librería MQTT (Publicación/Suscripción)
#include <NTPClient.h> //Librería para acceder a la hora actual mediante internet
#include <WiFiClientSecure.h>
#include <ESPmDNS.h>

const char* ssid = "*****"; //Indicar Red WIFI
const char* password = "*****"; // Contraseña red WIFI
const int mqtt_port = 8883; //Puerto Servidor MQTT SSL 8883
const char* mqtt_server = "192.168.1.39"; //Servidor MQTT EMQX
const char* mqtt_user = "595012"; //Usuario Servidor MQTT
const char* mqtt_pass = "*****"; //Contraseña Servidor MQTT

const char* cacert = \
    "-----BEGIN CERTIFICATE-----\n" \
    "MIIDZTCCAk2gAwIBAgIUUAUQAP3KMU/Bp8P13mrYdGZ24aQMwDQYJKoZIhvcNAQEL\n" \
    "BQAwwQjELMAkGA1UEBhMCQ04xIjAgBgNVBAoMGUVNVNUSBUZWNobm9sb2dpZXMgQ28u\n" \
    "LzCBMmGQxDzANBgNVBAMkVNUStYDQZAEw0yMjExMjYxMzYxMzYxMzYxMzYx\n" \
    "NjMyMzRAMEIxIjAgBgNVBAYTAkNOMSIwIAAYDQYJKoZIhvcNAQELAQYw\n" \
    "IENvLiwgTHRkMQ8wDQYDVQQDDAZFTVEgQ0EwggEiMA0GCSqGSIb3DQEBAQUAA4IB\n" \
    "DwAwggEKAoIBAQNDAoAobY62RUA7Bf31iwqhWriD36wfp6QBqryWe0UMwnVpcw\n" \
    "GDJUC47HDcx4URn7gzg+8HoZuc2LpAe9vAy8EXJuScX/9xXtP/3ukbrEf941muF\n" \
    "N0e10s/Sk60+6D+Bo1ZwtKKFFvDwNdgAhBUHELzWVH91HDC5TEs7GPiNXmqS4x\n" \
    "yE081hZGkKM81DUUJE7Cf+85h+QbtuZCpRe1/XIcSAXtHdkAChhhArqXy0Jkoza\n" \
    "vZcpnT5MdG0QQZtB4x/pJzY0tYuw3+0uNA8QKs+tySAaNLgykg+GLx0FRng5Qu2x\n" \
    "7WM6XKCEgqtCtyA4Iokv5HJQgp9zcS56WqSLAgMBAAGjUzBRMB0GA1UdDgQWBRRk\n" \
    "I/ps+CYtwquZc70vu2IMavBHejAFBgNVHSMGDAWgBRKI/ps+CYtwquZc70vu2IM\n" \
    "avBHejAPBgNVHRMBAf8EBTADAQH/MA0GCSqGSIb3DQEBCwUAA4IBAQAAnwqHVS85B\n" \
    "3EcdPwF9KHQ08qDUVCgtuIA1ZP8ImkqEq/sRvLjuWi8SEPavwKE/S8EE+22I6rPk\n" \
    "TyYkr2YjY4tnPrtVZxtPsSpA2DcmcYzy7vLUgYH5YhIvY4NYo9yM3TtCSfkfmeL\n" \
    "Awf9W4HF11S6a6LGuizDp3P0RXPxcGXubtDM6KH7N0LV+M2RR5pUIfzBcVwv953\n" \
    "wLtwABQ3h+QUkkY7xnZbFPYWTOCR0IcgIPbzZf4Jm/GgQexG370mzTLF+0df2HNW\n" \
    "AVqtFn7Bkg5o+AJyJUFwus1bLHL2du8i7XQ6cR8y2CC+51KtIh6E7ECL1WTMXHd6\n" \
    "MeCOBD7CnqGc\n" \
    "-----END CERTIFICATE-----";

const char *device_id = "home"; //Nombre del dispositivo
WiFiClientSecure espClient; //Crear cliente seguro que pueda conectarse a la IP y Puerto especificados
PubSubClient client(espClient); //Crear cliente de suscripción y publicación

//*****PINS*****
int rele_pin = 23; //Conexión del medidor de DUST
int co2;
int ubicacion;
int ventana;
bool ventana_anterior = false; //Estado inicial de la ventana

//Descargar la hora mediante internet
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 7200);

//*****PROGRAMA*****
void setup()
{
    Serial.begin(115200); //Configuración del puerto serie
    setup_wifi(); //Iniciar la conexión WIFI
    pinMode(rele_pin, INPUT); //Iniciar pin del sensor MICS-6814 para NO2

    espClient.setCACert(cacert);
    client.setServer(mqtt_server, mqtt_port); //Iniciar al servidor MQTT
    client.setCallback(callback); // Inicializar el bucle lectura payloads suscritos
    client.subscribe("home/ventana"); //Suscripción al topic ubicación
}
```

```

void setup_wifi()
{
  delay(10); //Esperar 0,01 segundos
  WiFi.begin(ssid, password); // Inicializar WIFI
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500); //Esperar 0,5 segundos
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println(ssid); //Imprimir al serial SSID
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); //Imprimir al serial IP
}

//Bucle lectura de los payload de los topics suscritos
void callback(char* topic, byte* payload, unsigned int length)
{
  String topicStr = topic; //Convertir a string para comparar topic
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.println(topicStr);
  char buff_p[length];

  if (topicStr == "595012/contaminante/co2") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String co2_c = String(buff_p); //Extraer payload como string
    co2 = co2_c.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/ubicacion") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String ubicacion_c = String(buff_p); //Extraer payload como string
    ubicacion = ubicacion_c.toInt(); //Convertir string a entero
  }

  if (topicStr == "595012/home/ventana") //Si el topic coincide, hacer lectura
  {
    for (int i = 0; i < length; i++)
    {
      buff_p[i] = (char)payload[i];
    }
    buff_p[length] = '\0'; //Seleccionar el último payload del topic
    String ventana_c = String(buff_p); //Extraer payload como string
    ventana = ventana_c.toInt(); //Convertir string a entero //tofloat??
  }
}

```



```

void reconnect()          // Bucle de reconexión
{
  while (!client.connected())    //Mientras el cliente no este conectado
  {
    Serial.print("Attempting MQTT connection...");
    if (client.connect(device_id, mqtt_user, mqtt_pass))//Conectar al client MQTT
    {
      Serial.println("connected");
      client.subscribe("home/ventana");          //Suscripciones
      delay(100);
    } else {
      Serial.print("fallo, rc=");
      Serial.print(client.state());
      Serial.println(" Intentar-lo en 5 segundos");
      delay(5000);          // Esperar 5 segundos antes de volver a intentar
    }
  }
}

void loop()    //Bucle principal
{
  rele();          //Leer bucle de sensor MQ135 para el CO2
  if (!client.connected())    //Si el cliente no se conecta
  {
    reconnect();          //Ir al bucle de reconnect
  }
  client.loop();
}

void rele()    //Bucle del sensor MQ135 para detectar el CO2
{
  if ((ubicacion == LOW) && (ventana = HIGH)) //Si detecta que esta en interior y hay ventana
  {
    if (co2 > 1000)
    {
      digitalWrite(rele_pin, HIGH); //Enciende el relé que controla la ventana para abrirla
      ventana_anterior = true;      //Guardar el estado de la ventana
    }
    else
    {
      digitalWrite(rele_pin, LOW); //Cierra el relé que controla la ventana para cerrarla
      ventana_anterior = false;     //Guardar el estado de la ventana
    }
  }
  else
  {
    digitalWrite(rele_pin, LOW); //Cierra el relé que controla la ventana para cerrarla
    ventana_anterior = false;     //Guardar el estado de la ventana
  }
}
}

```

## Anexo C. Código Docker, EMQX, InfluxDB, Telegraf y Grafana

```

version: "3"                                #Versión 3 Docker
services:                                    #Servicios que tendra el docker

  emqx:                                       #SERVICIO EMQX
    user: root                               #Debemos dar al usuario del servicio los permisos root
    image: emqx/emqx:5.0.4                   #Imagen de docker que utilizaremos en este caso EMQX

    environment:
      - EMQX_LISTENER__SSL__EXTERNAL__ACCEPTORS

    ports:                                    #Definir puerto que mapea y el puerto del contenedor
      - "18083:18083"                         #activar puerto Dashboard configuración EMQX
      - "1883:1883"                           #activar puerto 1883
      - "8883:8883"                           #activar puerto 8883 conexión segura
      - "8083:8083"
      - "8084:8084"

    volumes:
      - ./data/emqx/data:/opt/emqx/data      #Directorio de volumen de datos de data
      - ./data/emqx/log:/opt/emqx/log
      - ./data/emqx/etc/certs:/opt/emqx/etc/certs

  influxdb:                                   #SERVICIO InfluxDB
    user: root
    image: "influxdb:2.2.0-alpine"           #Imagen de docker que utilizaremos en este caso influxdb
    ports:
      - "8086:8086"                           #Puerto que utilizaremos de influxdb para conectar y ver que todo esta bien
    volumes:
      - ./data/influxdb/data:/var/lib/influxdb2 #Directorios de volumen de datos/persistencia necesarios de influxdb
      - ./data/influxdb/config:/etc/influxdb2
    environment:
      #Variables de entorno que se usara en influxdb
      - DOCKER_INFLUXDB_INIT_MODE=setup
      - DOCKER_INFLUXDB_INIT_USERNAME=*****
      - DOCKER_INFLUXDB_INIT_PASSWORD=*****
      - DOCKER_INFLUXDB_INIT_ORG=javiermontfe #Organización para almacenar los datos
      - DOCKER_INFLUXDB_INIT_BUCKET=datostfe
      - DOCKER_INFLUXDB_INIT_RETENTION=1w
      - DOCKER_INFLUXDB_INIT_ADMIN_TOKEN=javiermontfe-datostfe-auth-token #Token utilizado para conectar los diferentes servicios

  telegrafoutput:                             #Añadir nuevo servicio telegraf para enviar datos de EMQX a InfluxDB
    user: root
    image: "telegraf:1.22.4"
    deploy:
      restart_policy:
        condition: on-failure                 #Reiniciar servicio en caso de fallo
        delay: 10s                            #Cada 10 segundos durante 20 intentos
        max_attempts: 20
    volumes:
      - ./data/telegrafoutput/telegraf.conf:/etc/telegraf/telegraf.conf

  grafana:                                     #Servicio de GRAFANA
    user: root                                 #Hacer servicio como usuario root
    image: "grafana/grafana:8.5.3"           #Versión de GRAFANA
    ports:
      - "3000:3000"                           #Definir puerto de Grafana
    environment:
      - GF_SECURITY_ADMIN_PASSWORD__FILE=/run/secrets/admin_password #Leer contraseña de un archivo
    volumes:
      - ./data/grafana:/var/lib/grafana
    secrets:
      - source: grafana_admin_password
        target: /run/secrets/admin_password
  secrets:
    grafana_admin_password:
      file: ./secrets/grafana_admin_password #Archivo con la contraseña

```

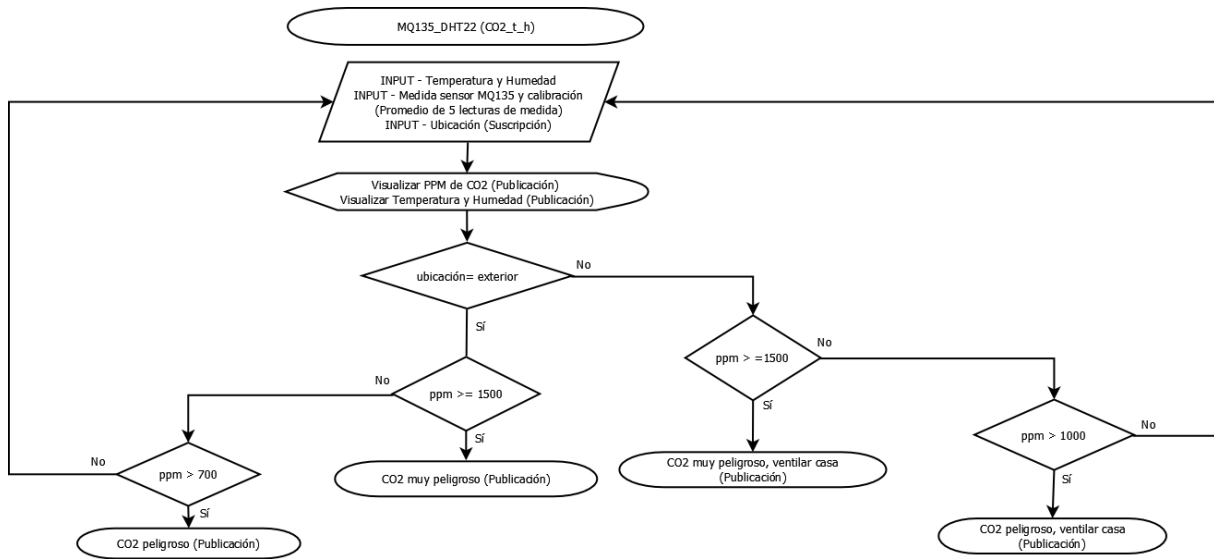
## Anexo D. Código Telegraf configuración

```
1 [global_tags]           #Etiquetas a nivel global
2   project = "TFE"
3
4 [agent]                 #Configuración del agente
5   interval = "10s"
6   round_interval = true
7   metric_batch_size = 1000
8   metric_buffer_limit = 10000
9   collection_jitter = "0s"
10  flush_interval = "10s"
11  flush_jitter = "0s"
12  precision = "0s"
13  hostname = "telegrafoutput"
14  omit_hostname = false
15
16 [[inputs.mqtt_consumer]] #Lee MQTT, en telegraf el que lee MQTT se llama consumer
17   servers = ["tcp://emqx:1883"] #Conectar al servidor
18
19   topics = [
20     "+/contaminante/#", "+/tiempo/#", "+/temperatura", "+/humedad"
21     #Y se genera en los topics concretos
22   ]
23
24   ## Username and password to connect MQTT server.
25   username = "influx"
26   password = "ix21a"
27
28   qos = 0
29   connection_timeout = "30s"
30   #qos = 2 #Calidad de servicio
31   data_format = "value"
32   data_type = "float"
33
34 [[outputs.influxdb_v2]] #Todos los datos leídos en el consumer ahora se van escribiendo en influx
35   urls = ["http://influxdb:8086"]
36   token = "javiermontfe-datostfe-auth-token" #Token anteriormente designado
37   organization = "javiermontfe" #Organización anteriormente designada
38   bucket = "datostfe" #Bucket anteriormente designado
```

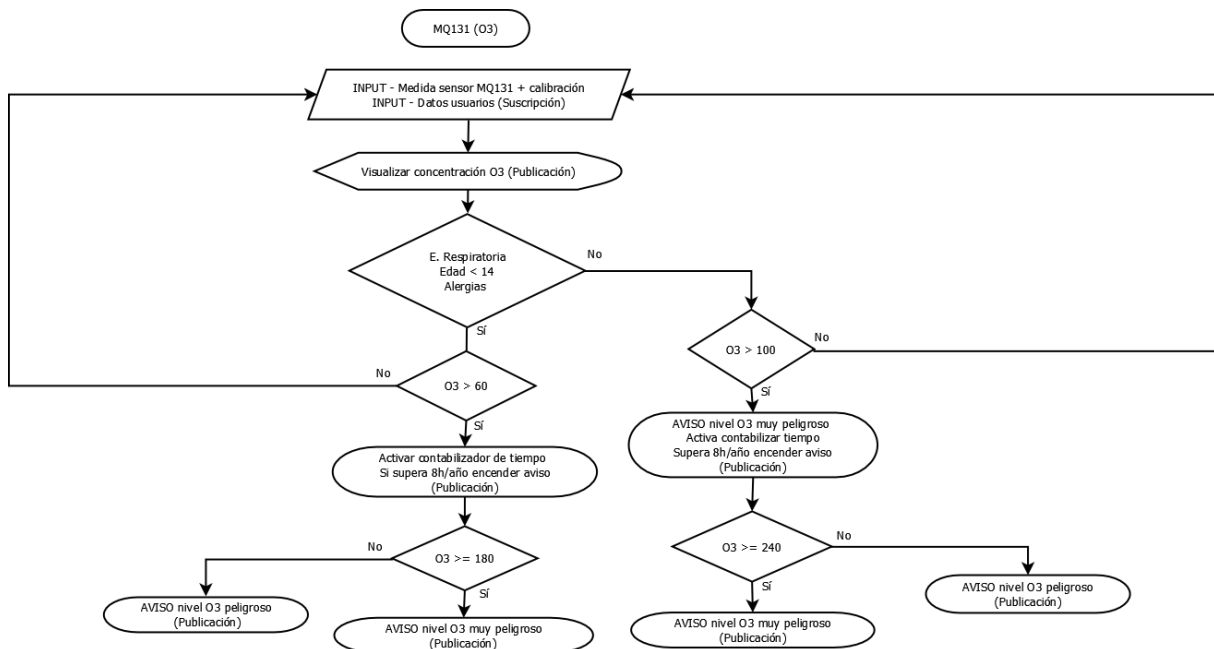


## Anexo E. Diagramas de funcionamiento

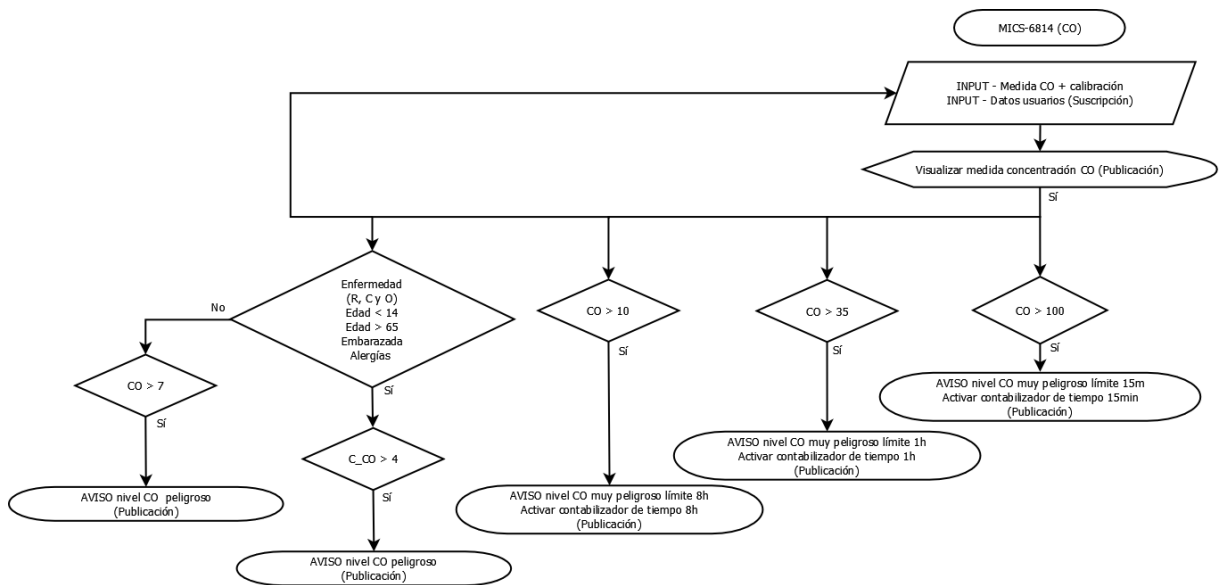
### Diagrama Funcionamiento MQ135 para CO<sub>2</sub>



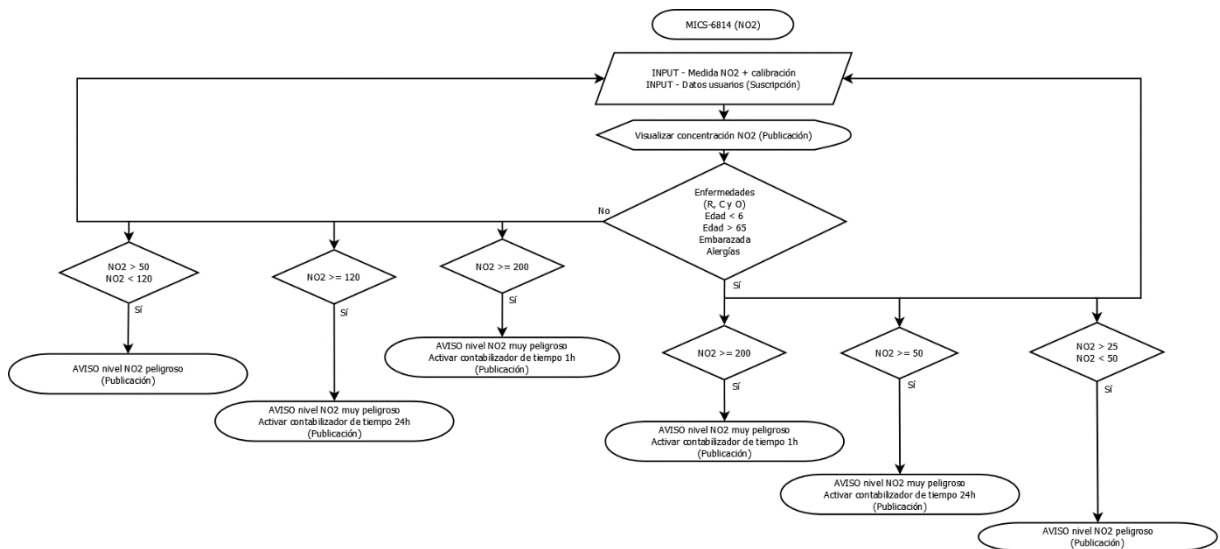
### Diagrama Funcionamiento MQ131 para O<sub>3</sub>



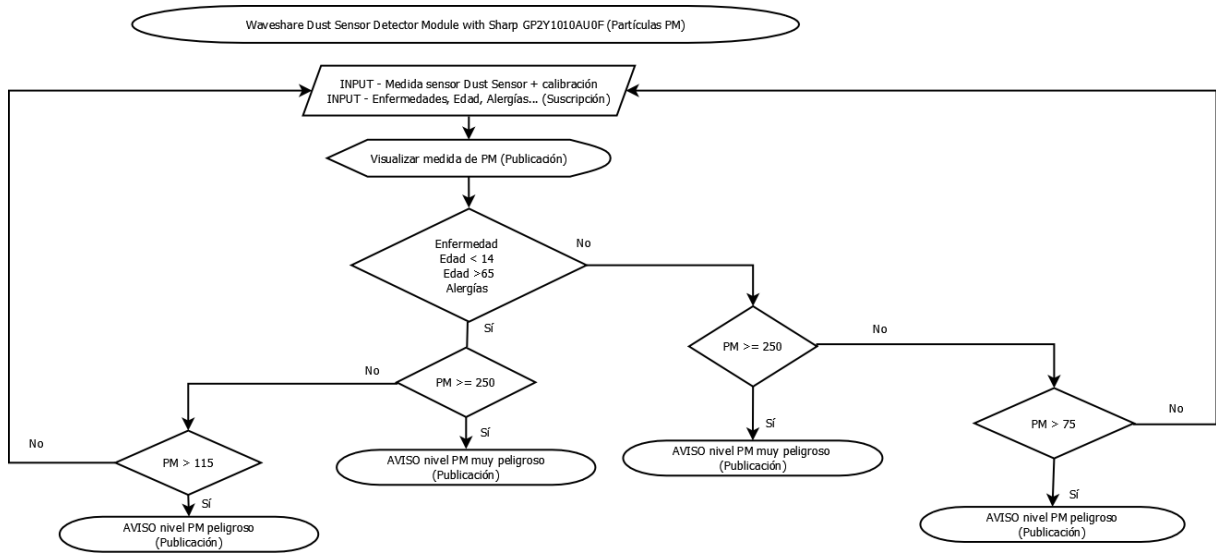
### Diagrama Funcionamiento MICS-6814 para CO



### Diagrama Funcionamiento MICS-6814 para NO<sub>2</sub>



### Diagrama Funcionamiento Sensor Dust para PM



## Anexo F. Aplicación propia Android Studio, SISCAS

### Gradle

```
apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
apply plugin: 'com.google.gms.google-services'

android {
    compileSdkVersion 33
    defaultConfig {
        applicationId "com.example.siscas"
        minSdkVersion 21
        targetSdkVersion 33
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
        }
    }
    namespace 'com.example.siscas'
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation "com.android.support:support-compat:28.0.0"
    implementation 'androidx.core:core-ktx:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.1.0'
    implementation 'org.eclipse.paho:org.eclipse.paho.android.service:1.1.1'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'org.chromium.net:cronet-embedded:98.4758.101'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'
    implementation 'com.github.Gruzer:simple-gauge-android:0.3.1'
    implementation 'pl.pawelkleczkowski.customgauge:CustomGauge:1.0.4'
    implementation 'org.bouncycastle:bcpkix-jdk15on:1.59'
    implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0'
    implementation 'org.influxdb:influxdb-java:2.21'
    implementation 'androidx.drawerlayout:drawerlayout:1.1.1'
    implementation 'androidx.coordinatorlayout:coordinatorlayout:1.2.0'
    implementation platform('com.google.firebase:firebase-bom:31.1.1')
    implementation 'com.google.firebase:firebase-analytics-ktx'
    implementation 'com.google.firebase:firebase-auth-ktx'
    implementation 'com.google.firebase:firebase-firestore-ktx'
    implementation 'com.google.firebase:firebase-database-ktx:20.1.0'
}
```

**AndroidManifest.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.POST_NOTIFICATIONS" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat"
        tools:targetApi="31">

        <activity
            android:name=".LoginActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
        <activity
            android:name=".MainActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
        </activity>
        <activity
            android:name=".RegistroActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
        </activity>
        <activity
            android:name=".TiemposContaminantesActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
        </activity>
        <activity
            android:name=".HelpActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
        </activity>
        <activity
            android:name=".RegistroLoginActivity"
            android:windowSoftInputMode="adjustPan"
            android:theme="@style/AppTheme.NoActionBar"
            android:exported="true">
        </activity>

        <service android:name="org.eclipse.paho.android.service.MqttService" />
    </application>
</manifest>

```

## Contaminación Layout

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="MainActivity"
    tools:ignore="ExtraText"
    tools:viewBindingIgnore="true">

    <com.ekn.gruzer.gaugelibrary.HalfGauge
        android:id="@+id/CO2gauge"
        android:layout_width="173dp"
        android:layout_height="179dp"
        android:layout_marginTop="98dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.927"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <TextView
        android:id="@+id/textCO2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="220dp"
        android:fontFamily="@font/allertastencilregular"
        android:text="@string/co2"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/CO2gauge"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.417"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textO3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:layout_marginBottom="140dp"
        android:fontFamily="@font/allertastencilregular"
        android:text="O3"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/O3gauge"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textPM"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="224dp"
        android:layout_marginBottom="140dp"
        android:fontFamily="@font/allertastencilregular"
        android:text="PM"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/PMgauge"
        app:layout_constraintStart_toStartOf="parent"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textCO"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="20dp"
        android:fontFamily="@font/allertastencilregular"
        android:text="CO"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/COgauge"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.668"
        tools:ignore="MissingConstraints" />

    <TextView
        android:id="@+id/textNO2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="216dp"
        android:fontFamily="@font/allertastencilregular"
        android:text="NO2"
        android:textSize="20sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toBottomOf="@+id/COgauge"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.685"
        tools:ignore="MissingConstraints" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

```
<com.ekn.gruzer.gaugelibrary.HalfGauge
    android:id="@+id/PMgauge"
    android:layout_width="173dp"
    android:layout_height="179dp"
    android:layout_marginTop="468dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.927"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<com.ekn.gruzer.gaugelibrary.HalfGauge
    android:id="@+id/O3gauge"
    android:layout_width="173dp"
    android:layout_height="179dp"
    android:layout_marginTop="468dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.072"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<com.ekn.gruzer.gaugelibrary.HalfGauge
    android:id="@+id/COgauge"
    android:layout_width="173dp"
    android:layout_height="179dp"
    android:layout_marginTop="280dp"
    android:layout_marginBottom="9dp"
    app:layout_constraintBottom_toBottomOf="@+id/O3gauge"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.072"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.0" />

<com.ekn.gruzer.gaugelibrary.HalfGauge
    android:id="@+id/NO2gauge"
    android:layout_width="173dp"
    android:layout_height="179dp"
    android:layout_marginTop="280dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.927"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<Switch
    android:id="@+id/switch1"
    android:layout_width="140dp"
    android:layout_height="45dp"
    android:layout_marginTop="48dp"
    android:minHeight="48dp"
    android:text="@string/switch1"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.151"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/humedad"
    tools:ignore="MissingConstraints,UseSwitchCompatOrMaterialXml" />

<TextView
    android:id="@+id/temperatura"
    android:layout_width="200dp"
    android:layout_height="30dp"
    android:layout_marginTop="88dp"
    android:layout_marginEnd="192dp"
    android:drawableStart="@drawable/thermometer"
    android:drawablePadding="16dp"
    android:fontFamily="@font/staatliches"
    android:gravity="bottom"
    android:paddingStart="5dp"
    android:scrollbars="vertical"
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    android:textColor="#000000"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.804"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/humedad"
    android:layout_width="180dp"
    android:layout_height="30dp"
    android:layout_marginTop="128dp"
    android:layout_marginEnd="280dp"
    android:drawableStart="@drawable/water_percent"
    android:drawablePadding="16dp"
    android:fontFamily="@font/staatliches"
    android:gravity="bottom"
    android:paddingStart="5dp"
    android:scrollbars="vertical"
    android:textAppearance="@style/TextAppearance.AppCompat.Medium"
    android:textColor="#000000"
    android:textSize="18sp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.804"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

## CLIENTE MQTT – Ejemplo en MainActivity

```

class ClienteMQTT_MainActivity(context: MainActivity) {
    private var mqttClient: MqttAndroidClient

    companion object {
        const val TAG = "AndroidMqttClient"
    }

    fun setCallback(callback: MqttCallbackExtended?) {
        mqttClient.setCallback(callback)
    }

    init {
        mqttClient = MqttAndroidClient(context, MQTT_BROKER, "595012app")
        val options = MqttConnectOptions()
        val sslSocketFactory = SSLSocketFactory.getDefault() //Seguridad SSL
        options.socketFactory = sslSocketFactory
        options.isAutomaticReconnect = true
        options.isCleanSession = true
        options.userName = MQTT_USER
        options.password = MQTT_PASS.toCharArray()
        options.connectionTimeout = 10
        options.keepAliveInterval = 60
        try {
            val caCrtFile = context!!.resources.openRawResource(R.raw.ca)
            options.socketFactory = getSingleSocketFactory(caCrtFile)
            mqttClient.connect(options, null, object : IMqttActionListener {
                override fun onSuccess(asyncActionToken: IMqttToken?) {
                    Log.d(TAG, "Connection success")
                    suscripcionTopics()
                }
                override fun onFailure(asyncActionToken: IMqttToken?, exception: Throwable?) {
                    Log.d(TAG, "Connection failure")
                }
            })
        } catch (e: MqttException) {
            e.printStackTrace()
        }
    }

    fun publish(topic: String, msg: String, qos: Int = 1, retained: Boolean = false) {
        try {
            val message = MqttMessage()
            message.payload = msg.toByteArray()
            message.qos = qos
            message.isRetained = retained
            mqttClient.publish(topic, message, null, object : IMqttActionListener {
                override fun onSuccess(asyncActionToken: IMqttToken?) {
                    Log.d(TAG, "$msg published to $topic")
                }
                override fun onFailure(asyncActionToken: IMqttToken?, exception: Throwable?) {
                    Log.d(TAG, "Failed to publish $msg to $topic")
                }
            })
        } catch (e: MqttException) {
            e.printStackTrace()
        }
    }

    private fun suscripcionTopics(){
        try {
            mqttClient.subscribe("595012/temperatura",1)
            mqttClient.subscribe("595012/humedad",1)
            mqttClient.subscribe("595012/contaminante/co2",1)
            mqttClient.subscribe("595012/contaminante/co",1)
            mqttClient.subscribe("595012/contaminante/no2",1)
            mqttClient.subscribe("595012/contaminante/ozono",1)
            mqttClient.subscribe("595012/contaminante/pm",1)
            mqttClient.subscribe("595012/aviso/co2",1)
            mqttClient.subscribe("595012/aviso/co",1)
            mqttClient.subscribe("595012/aviso/no2",1)
            mqttClient.subscribe("595012/aviso/ozono",1)
        }catch (e: MqttException){
            e.printStackTrace()
        }
    }

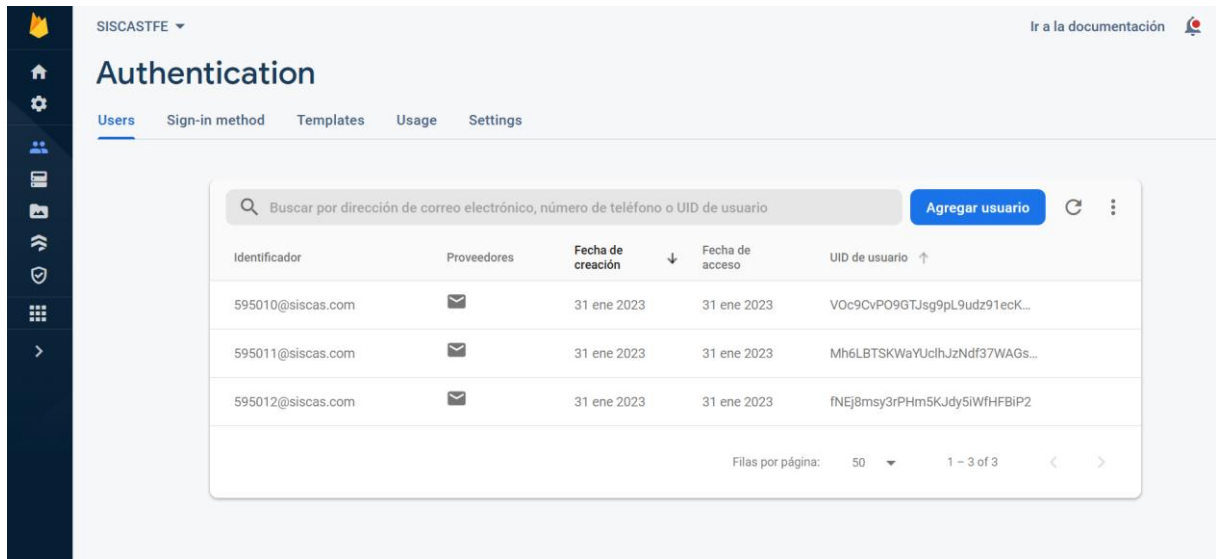
    //Vuelve true si esta conectado
    fun isConnected(): Boolean {
        return mqttClient.isConnected
    }

    fun destroy() {
        mqttClient.unregisterResources()
        mqttClient.disconnect()
    }

    @Throws(Exception::class)
    fun getSingleSocketFactory(caCrtFileInputStream: InputStream?): SSLSocketFactory? {
        Security.addProvider(BouncyCastleProvider())
        var caCert: X509Certificate? = null
        val bis = BufferedInputStream(caCrtFileInputStream)
        val cf = CertificateFactory.getInstance("X.509")
        while (bis.available() > 0) {
            caCert = cf.generateCertificate(bis) as X509Certificate
        }
        val caKs = KeyStore.getInstance(KeyStore.getDefaultType())
        caKs.load(null, null)
        caKs.setCertificateEntry("cert-certificate", caCert)
        val tmf = TrustManagerFactory.getInstance(TrustManagerFactory.getDefaultAlgorithm())
        tmf.init(caKs)
        val sslContext = SSLContext.getInstance("TLSv1.2")
        sslContext.init(null, tmf.trustManagers, null)
        return sslContext.socketFactory
    }
}

```

## Firestore Authentication



## Firestore Realtime Database

