



Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias de la Electrónica
Maestría en Ingeniería Electrónica,
opción Instrumentación Electrónica

Tesis para obtener el grado de
Maestro en ingeniería Electrónica

DESARROLLO DE UN SISTEMA EMBEBIDO PARA
ADVERTIR SOBRE LAS CONDICIONES DE RIESGO DE
CONTAGIO DE COVID-19 MEDIANTE EL MONITOREO
DE LA CALIDAD DEL AIRE

Presenta:
Lic. Yair Romero López*

Asesor:
M.C. Ricardo Álvarez González

Coasesor:
M.C. Rodrigo Lucio Maya Ramírez

Asesora externa:
M.C. Alba Maribel Sánchez Gálvez

“Como no estás experimentado en las cosas del mundo, todas las cosas que tienen algo de dificultad te parecen imposibles. Confía en el tiempo que suele dar dulces salidas a muchas amargas dificultades”

-Don Quijote de la Mancha, Miguel de Cervantes Saavedra.

Agradecimientos

A CONAHCYT por la beca otorgada durante dos años de duración de mi maestría.

A CONCYTEP por el apoyo otorgado durante mi estancia académica en el extranjero.

A la BUAP mi más profundo agradecimiento por haber sido mi alma mater y por haber enriquecido mi vida de manera inimaginable me ha proporcionado no solo conocimiento académico, sino también valores fundamentales, habilidades de resolución de problemas y un sentido de comunidad que me han preparado para enfrentar los desafíos del mundo real.

Dedicatoria

Agradezco profundamente a mis padres Carlos Romero Hernández y María Petra López Parra, por su inquebrantable dedicación, incansable esfuerzo y constante apoyo al proporcionarme todas las herramientas indispensables para lograr cumplir mis sueños y metas. Mi gratitud se dirige a mi madre, quien no solo me escuchó, sino que también me alentó incansablemente a superarme. A mi padre, mi mayor fuente de motivación y un modelo a seguir insuperable. Gracias por el ejemplo a seguir y por moldear mi carácter de una manera que me enorgullece.

A mi abuelo Carlos Roman Romero Lima, quien ha tenido fe en las cosas que hago, posiblemente el mayor admirador de la ciencia que he elegido, y que a través de mis proyectos muestra la pasión que compartimos por el saber.

A mi hermano Carlos Uriel Romero López, quien ha sido un confidente y consejero, tu dedicación a mi éxito y tu inquebrantable fe en mí han sido un viento favorable en las velas de mi determinación en este largo camino de éxito profesional.

A mis sobrinos Carlos Alonso e Ian Osiel Romero Gómez, quienes les dedico este trabajo, llenan mi corazón de alegría. Ustedes son un recordatorio constante de la belleza y la inocencia que el mundo puede ofrecer. Su curiosidad inagotable y la capacidad para ver el mundo con ojos asombrados me inspira a redescubrir la magia en las cosas más simples. A medida que crecen y exploran, espero que nunca pierdan ese sentido innato de asombro y admiración por todo lo que les rodea.

A mi pareja Abigail Rosas Maldonado, por su paciencia infinita, comprensión y apoyo emocional, no tengo palabras suficientes para expresar mi gratitud. Tu presencia en mi vida ha sido mi refugio y fuente de inspiración. Las dificultades que hemos pasado a lo largo de nuestra vida profesional, se han convertido en los cimientos sobre los cuales construimos nuestro camino. En lugar de desalentarnos, nos han fortalecido y motivado a persistir, a ser mejores y a avanzar con determinación en cada desafío. Esa capacidad de enfrentar juntos las dificultades nos guiará hacia logros aún más grandiosos. Contigo, he aprendido que el amor y el apoyo mutuo no solo hacen que los momentos felices sean más brillantes,

sino que también convierten las pruebas en oportunidades para crecer y florecer.

Estoy agradecido con mis asesores los Maestro Ricardo Álvarez González, Rodrigo Lucio Maya Ramírez y Alba Maribel Sánchez Gálvez, por su constante disposición para discutir ideas, el apoyo mostrado en mis aportes y sus consejos han enriquecido el contenido de esta tesis y me han animado a abordar los desafíos con confianza. Su experiencia ha sido fundamental para la profundidad y solidez de este trabajo.

Agradezco profundamente a mi asesor de estancia académica el Doctor Alejandro Paredes Camacho, quien brindó un invaluable apoyo en la ejecución de un proyecto de internacionalización. Su dedicación de tiempo y esfuerzo para enriquecer esta investigación ha dejado una huella significativa en mi desarrollo profesional. Además, su ejemplo y compromiso, en un contexto cultural distinto, han sido una fuente de inspiración y orgullo.

Quiero expresar mi sincero agradecimiento a la coordinación de mi maestría y al cuerpo académico de la misma, por su valioso apoyo y orientación a lo largo de este proceso de investigación. Su compromiso con la excelencia académica y su dedicación para facilitar el desarrollo de nuestros conocimientos han sido fundamentales en mi crecimiento como estudiante.

Al honorable jurado quien enriqueció esta tesis, y sus críticas han sido constructivas buscando el mejoramiento de este proyecto.

No puedo dejar de mencionar a mis compañeros y amigos de maestría, quienes han sido consejeros y maestros en este trayecto de dos años. Las conversaciones motivadoras y las tardes compartidas, son un recuerdo que llevaré en el corazón.

Índice general

1. Introducción	12
1.1. Objetivo general	13
1.2. Objetivos específicos	13
1.3. Justificación	13
1.4. Descripción del sistema	14
1.4.1. Sistema embebido	15
1.4.2. Recepción y almacenamiento de datos	16
1.4.3. Obtención y visualización de datos mediante un <i>dashboard</i>	16
1.5. Estado del arte	16
1.5.1. Sistemas embebidos existentes en el mercado, para el monitoreo de CO ₂	17
1.5.2. Comparativa con proyectos similares	19
1.5.3. Conclusiones del estado del arte	20
2. Marco teórico	22
2.1. Sistema embebido	25
2.1.1. Diseño de placas de circuito impreso (PCB) para sistemas embebidos	26
2.1.2. <i>Software</i> de desarrollo para el diseño de PCB: <i>Eagle</i>	27
2.2. Sensores para el monitoreo de la calidad del aire y sus parámetros	28
2.2.1. Principio operativo de los sensores infrarrojos NDIR	30
2.2.2. Sensor <i>SenseAir S8</i>	32
2.3. Microcontroladores y SoC	35
2.3.1. ESP-32	39
2.4. <i>Python</i> para microcontroladores: <i>MicroPython</i>	42
2.4.1. Thonny IDE	44
2.5. Comunicación inalámbrica y acceso a los datos	45
2.5.1. Protocolo de comunicación MQTT	47
2.5.2. Características principales de MQTT	48
2.6. Plataforma <i>IoT ThingSpeak</i>	49
2.6.1. Características principales de <i>ThingSpeak</i>	49

3. Desarrollo	51
3.1. Desarrollo del <i>hardware</i>	51
3.1.1. Pruebas de funcionamiento en placa de prototipo electrónico	52
3.1.2. Diseño de PCB ESP-32 WROOM en una placa base	54
3.1.3. ECAD de la ESP32-WROOM en una placa base .	57
3.1.4. Fabricación y ensamblaje de PCB ESP32 WROOM	59
3.2. Programación del <i>firmware</i> de la placa base ESP32-WROOM	61
3.3. Desarrollo del <i>firmware</i> de la placa base ESP32-WROOM	62
3.4. Desarrollo de una <i>dashboard</i> en <i>software IoT: ThingSpeak</i>	65
3.5. Pruebas de conexión hacia la nube	65
3.5.1. Creación de un tablero en el <i>software ThingSpeak</i>	65
3.6. Versión final del <i>hardware</i> del sistema integrado.	67
3.6.1. Módulo convertidor AC-DC:	68
3.6.2. Cargador de baterías de tecnología LiPo	69
3.6.3. Baterías 18650	71
3.6.4. Convertidor elevador de voltaje	72
3.7. Implementación de la versión final del sistema embebido en placa de prototipo electrónico	74
3.8. Pruebas de funcionamiento del sistema embebido en placa de prototipos electrónicos, implementando un sensor de tecnología NDIR	75
3.8.1. Protocolo de comunicación sensor SenseAir S8	77
3.9. Diseño de PCB del sistema embebido	80
3.10. Diseño ECAD para la implementación de la estructura externa de protección del sistema embebido	84
3.11. Adición de sensores HVAC	87
3.11.1. Sensor BME280	88
3.12. Prototipo final	90
3.13. Implementación y recolección de datos en el <i>software IoT</i> en <i>Thingspeak</i>	92
3.14. Implementación recolección de datos en una aplicación para <i>smartphone</i> con sistema operativo Android	93
3.15. Generador de tweets por el sistema embebido	94
3.16. Aplicación del sistema embebido	97
4. Análisis de datos	101

4.1.	Estado del arte	101
4.2.	Power BI data y desarrollo de una <i>dashboard</i>	103
4.3.	Adición de sensado de nuevas variables	106
4.4.	Desarrollo de algoritmos de análisis de datos	108
4.5.	Predicciones automatizadas con <i>Power BI</i>	108
4.6.	Modelo ARIMA	109
4.7.	Predicción de niveles de CO ₂ en interiores	112
4.8.	Correlación de las variables medidas: CO ₂ , temperatura, humedad relativa y presión atmosférica	113
4.9.	Conclusiones del análisis de datos	118
	Conclusiones	121
	5. Bibliografía	123
	6. Apéndices	131
6.1.	Esquemáticos diseñados	131
6.2.	Materiales usados para el diseño de la placa electrónica .	134
6.3.	Algoritmos realizados en <i>MicroPython</i>	135
6.3.1.	Programa para la comunicación serial del Senseair S8 y el ESP-32	135
6.3.2.	Programa para el uso del sensor BME280	135
6.3.3.	Programa para el envío de notificaciones vía <i>Twitter</i>	143

Índice de figuras

1.1.	Diagrama general del sistema.	15
1.2.	Monitor de CO ₂ comercial.	17
2.1.	Técnicas de mejora para la calidad del aire interior (IAQ) para el combate del COVID-19.	23
2.2.	Diagrama general de los sistemas <i>IoT</i>	24
2.3.	Componentes de los sistemas embebidos.	26
2.4.	<i>Software</i> comúnmente usado en el diseño de placas electrónicas.	27
2.5.	Integración de los <i>software Eagle y Fusion 360</i>	28
2.6.	Sensor NDIR Mhz19-B.	30
2.7.	Principio del sensor de gas NDIR.	31
2.8.	<i>SenseaAir S8</i>	33
2.9.	Módulo ESP 32.	36
2.10.	Arduino Uno <i>WiFi</i>	37
2.11.	Raspberry pi 4 Modelo B.	38
2.12.	Esp-32 seccionado.	40
2.13.	Ambiente gráfico del <i>software Thonny</i>	45
2.14.	Dashboard presentado en el <i>software ThingSpeak</i>	47
2.15.	Modelo de comunicación MQTT.	48
3.1.	Diagrama de bloques del desarrollo del sistema.	51
3.2.	Prototipo sistema embebido.	53
3.3.	Diseño de la PCB ESP-32 en una placa base, izquierda: vista superior, derecha: vista inferior.	57
3.4.	Colocación de componentes electrónicos modelados en 3D.	57
3.5.	ECAD desarrollado en el <i>software Fusion 360</i> del ESP32-WROOM en una placa base.	59
3.6.	Imágenes generadas de la placa electrónica en 3D.	59
3.7.	PCB sin ensamblar.	60
3.8.	PCB ensamblada.	61
3.9.	Ventanas generadas para la instalación del <i>firmware de MicroPython</i> en el <i>software Thonny</i>	62
3.10.	Algoritmo implementado en el ESP-32.	64
3.11.	Recursos usados para pruebas de conexión a Internet.	65

3.12. Canal generado para las primeras pruebas de conexión hacia la nube.	66
3.13. <i>Dashboard</i> generada en el <i>software ThingSpeak</i>	66
3.14. Diagrama de bloques de la versión final del sistema embebido	68
3.15. Esquema de implementación del módulo de conversión de AC-DC	69
3.16. Esquema de implementación del circuito cargador de baterías y su etapa de protección	70
3.17. Batería 18650 usada en el sistema embebido	72
3.18. Esquema de implementación del módulo de conversión de AC-DC.	74
3.19. Prueba de la versión final del sistema embebido en placa de prototipos electrónicos.	75
3.20. Sensor NDIR SenseAir S8.	76
3.21. Comunicación serial entre el Módulo ESP-32 y el SenseAir S8.	77
3.22. Comunicación serial entre el Módulo ESP-32 y el SenseAir S8.	79
3.23. Versión final de la PCB del sistema embebido.	81
3.24. Visualización de la PCB y sus respectivos bloques.	84
3.25. Renderizado de la versión final de la PCB.	85
3.26. Renderizado del producto final del sistema embebido.	86
3.27. Renderizado del sensor BME280.	89
3.28. Colocación del sensor.	90
3.29. Visualización de los archivos STL en el <i>software Ultimate Cura</i>	91
3.30. Producto final del sistema embebido.	91
3.31. Visualización de las variables medidas en el <i>software Thingspeak</i>	92
3.32. Presentación de la <i>dashboard</i> visualizada en el aplicación móvil.	94
3.33. Visualización del <i>Tweet</i> generado por si el sistema embebido al alcanzar un nivel de CO ₂ > 1000 ppm.	95
3.34. Algoritmo final del <i>firmware</i> implementado en el sistema embebido.	96
3.35. Imágenes mostradas en la pantalla del sistema embebido.	97
3.36. Salon de pruebas para el monitoreo del CO ₂	98

3.37. Salón de pruebas para el monitoreo del CO ₂	98
3.38. Salón de pruebas para el monitoreo del CO ₂	100
4.1. Página 1 de la <i>dashboard</i> generada en <i>Power BI</i>	105
4.2. Página 2 de la <i>Dashboard</i> generada en <i>Power BI</i>	106
4.3. Colocación del sensor añadido.	107
4.4. Visualización de las imágenes mostradas en la pantalla del sistema embebido.	108
4.5. Predicción de valores de CO ₂ en <i>Power BI</i>	109
4.6. Algoritmo ARIMA implementado para la predicción de valores de CO ₂	110
4.7. Comparación de valores reales y pronosticados de CO ₂	111
4.8. Diagrama de bloques con la adición del análisis de datos.	113
4.9. Algoritmo de regresión lineal implementado para la predicción de valores de CO ₂	115
4.10. Matriz de correlación entre variables.	116
4.11. Matriz de correlación entre variables.	117
4.12. Comparación de los valores reales vs pronosticados.	118
6.1. Esquemático del sistema prueba montado en protoboard.	131
6.2. Esquemático ESP32 WROOM en placa base.	132
6.3. Esquemático final del sistema embebido.	133

Índice de tablas

1.1. Comparativa de los sistemas de monitoreo existentes en el mercado.	18
2.1. <i>Hardware</i> y <i>software</i> comúnmente usado en la elaboración de trabajos de <i>IoT</i>	25
2.2. Especificaciones más notables del <i>SenseAir S8</i>	33
2.3. Variaciones del <i>SenseAir S8</i>	35
2.4. Especificaciones técnicas ESP32 WROOM.	40
2.5. Dispositivos más populares usados para el uso de <i>MicroPython</i>	43
3.1. Formato de tabla generada por las mediciones obtenidas	67
3.2. Estado de los indicadores leds en función de carga de la batería	71
3.3. Especificaciones nominales de la batería 18650.	72
3.4. Especificaciones usadas para la comunicación con el sensor Senseair S8.	78
3.5. Cadena de Bits enviados y recibidos por el sensor Senseair S8.	78
3.6. Opciones para exportar los datos almacenados en <i>ThingSpeak</i>	93
4.1. Algoritmo ARIMA implementado para la predicción de valores de CO ₂	111

1. Introducción

A principios de diciembre de 2019, se detectó una neumonía de origen desconocido en la ciudad de Wuhan (China). A raíz de ello, las autoridades sanitarias de China se vieron sorprendidas por una serie de neumonías de origen desconocido que poseía una gran facilidad para su expansión. Al virus causante, perteneciente a la familia Coronarividae, se le denominó coronavirus 2 del síndrome respiratorio agudo grave (SARS-CoV-2), y a la enfermedad, COVID-19[1].

En un principio se pensó que el brote epidémico podría ser controlado a nivel local en China. El 11 de marzo de 2020, ante la rápida y progresiva expansión de la epidemia a nivel internacional, la OMS decretó el estado de pandemia.[2]

La enfermedad fue transportada rápidamente a escala global por individuos inicialmente sin síntomas viajando desde China a Europa y Estados Unidos principalmente. En México se detectó por primera vez en febrero del 2020 en un individuo que estuvo de viaje en Italia [3].

El impacto súbito y generalizado de la pandemia del COVID-19, ha afectado a las actividades esenciales y presenciales de nuestra vida cotidiana durante casi dos años, a pesar de la existencia de nuevas variantes del virus, existe un número significativo de la población vacunada en nuestro país, esto ha dado origen a la propuesta del retorno a las actividades presenciales, en sectores educativos, turísticos, profesionales, etc.

Por ello se requiere todas las medidas de protección e higiene, además de una planificación muy estricta, para minimizar el riesgo de contagio. La medición de la concentración de dióxido de carbono (CO_2) es una estrategia que puede advertir el riesgo de contagio de la enfermedad del coronavirus (COVID-19) en un espacio cerrado donde se encuentre reunido un grupo de personas. El resultado puede proporcionar información, a partir de la cual se puede deducir si la ventilación es adecuada o deficiente, lo que facilitaría la propagación del virus.

1.1. Objetivo general

Desarrollar un sistema embebido para monitorear la concentración de CO₂ en un espacio cerrado y emitir alarmas cuando se excedan los parámetros establecidos, adicionalmente llevar un registro diario de los datos, almacenándolos en la nube.

1.2. Objetivos específicos

1. Determinar las características que deben tener los dispositivos del sistema embebido.
2. Diseñar un sistema embebido para el envío de datos procesados a través del sensor de CO₂.
3. Estimar niveles de CO₂ óptimos para la convivencia, en caso de no ser así, notificarlo a las personas que se encuentran en el recinto.
4. Enviar y almacenar en la nube, la información sobre los niveles de CO₂ registrados durante el día.
5. Integrar y verificar el funcionamiento del sistema embebido y la conexión de una plataforma *IoT* (*Internet of things*) para la visualización de los datos obtenidos.

1.3. Justificación

Actualmente la OMS plantea que el COVID-19, se transmite principalmente a través de microgotas. La inoculación de microgotas en las vías respiratorias deviene de la exposición del hospedador a eventos del paciente (tos, estornudo, carraspeo, etc.) o procedimientos que inducen dispersión de gotitas en el aire. Para la OMS la mayoría de los contagios se producen a través del contacto cercano, motivo por el cual el distanciamiento social debe ser de 1 m. [4]

Se tiene el conocimiento generado por estudios de brotes de enfermedades respiratorias, estudios de dinámica de partículas y modelos matemáticos que este virus ingresa mediante la respiración de los individuos, a corta distancia entre personas existen partículas al respirar y al

hablar las cuales se denominan gotículas y aerosoles lo cual; en espacios cerrados provoca eventos de súper-diseminación del virus. [5][6]

La ventilación y la limpieza del aire ahora se reconocen como herramientas importantes para mitigar el riesgo de COVID-19 en interiores. Sin embargo, muchos espacios públicos están mal ventilados o su estado de ventilación puede ser desconocido para los usuarios. Aunque se han desarrollado varias calculadoras de ventilación para ayudar a evaluar las necesidades de ventilación, estas herramientas aún requieren una cierta cantidad de datos y conocimientos técnicos para su implementación.

El monitoreo de CO₂ es un medio establecido para evaluar si la ventilación es adecuada para el número de personas que ocupan el espacio. Aunque los niveles de CO₂ oscilan entre 350 y 450 ppm al aire libre, las personas que se reúnen y respiran dentro de un edificio harán que el CO₂ acumule a niveles mucho más altos al menos que se elimine a través de la ventilación. Cuantas más personas ocupen un espacio y más intensa sea su actividad física, más ventilación se requiere para mantener la comodidad de los ocupantes.

Además, la cantidad de personas y la naturaleza de sus actividades dentro de un espacio pueden afectar la calidad del aire y la presencia de partículas infecciosas a lo largo del tiempo. Se ha propuesto en el presente trabajo, medir las concentraciones de CO₂ en interiores utilizando sensores de bajo costo como una forma de estimar el riesgo de transmisión de COVID-19 en interiores. [7]

El proyecto que se llevará a cabo pretende ser una referencia para el desarrollo de un sistema de monitoreo y alertamiento para advertir sobre las condiciones de riesgo de contagio del COVID-19, esto podría generar un impacto científico importante, ya que los datos y la información generada podrá emplearse por otros colegas para desarrollar sistemas semejantes y/o para realizar una trazabilidad y controlar la propagación de la infección y, de este modo, continuar el desarrollo tecnológico para el beneficio de la sociedad.

1.4. Descripción del sistema

El proyecto está dividido en 3 bloques fundamentales los cuales se describen a continuación:

- Sistema embebido.
- Recepción y almacenamiento de datos.
- *Dashboard*.

En la figura 1.1 se muestran los componentes de cada bloque.

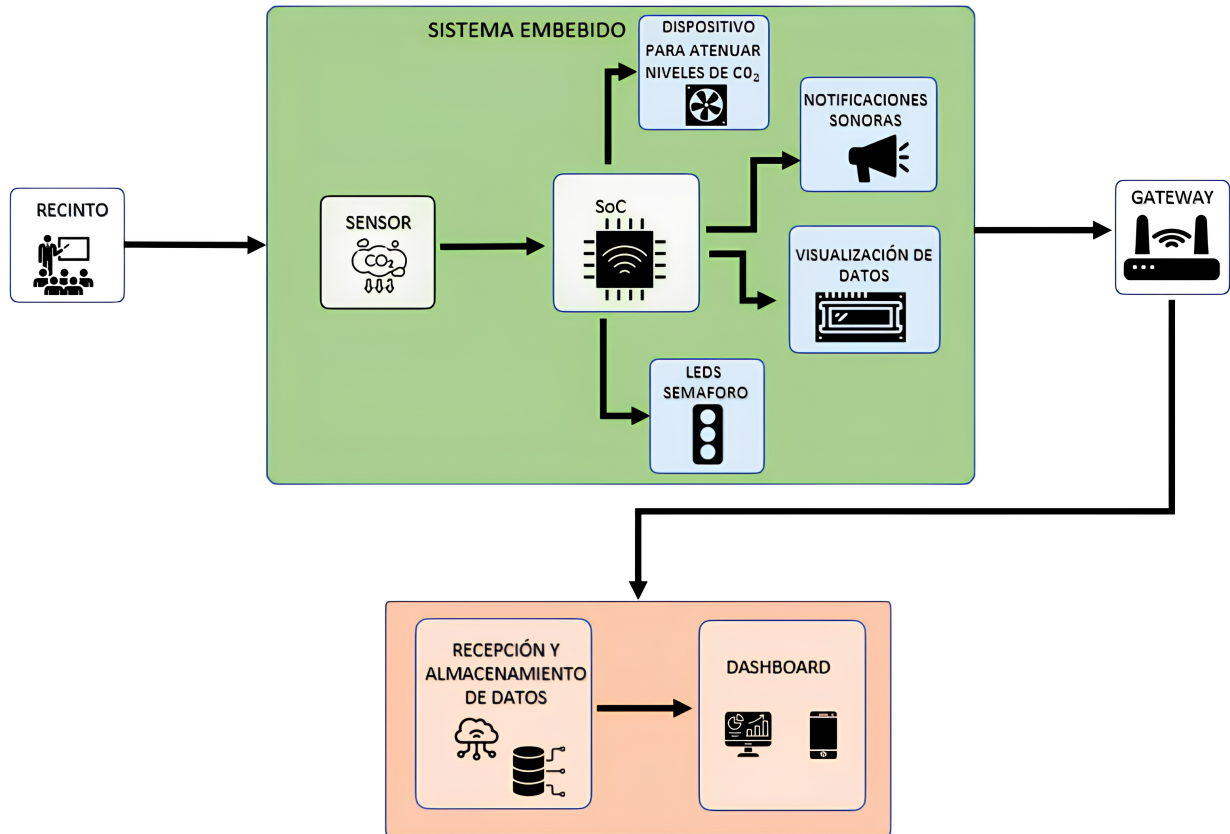


Figura 1.1: Diagrama general del sistema.

1.4.1. Sistema embebido

El sistema embebido se divide en 4 bloques fundamentales, los cuales se describen a continuación:

- Sensor de CO₂: Módulo que incluye un sensor de gas el cual detectará las partes por millón del CO₂ en el ambiente.
- SoC: Sistema contenido en un solo chip, donde se encontrará el microcontrolador y el *firmware* correspondiente, al igual que la respectiva etapa electrónica para el envío de información inalámbrica vía *Wi-Fi*.

- Pantalla electrónica: Dispositivo el cual notificará de forma visual los datos obtenidos a través del sensor.
- Salida de activación electrónica: Se generará una señal de control, con la finalidad de activar algún dispositivo que mitigue los niveles de CO₂ en el ambiente.

1.4.2. Recepción y almacenamiento de datos

Se hará uso de una plataforma *IoT* ya existente, la cual se conectará con el sistema embebido mediante un protocolo de comunicación inalámbrica, de este modo, se llevará a cabo la recepción y almacenamiento de los datos obtenidos.

1.4.3. Obtención y visualización de datos mediante un *dashboard*

Los datos obtenidos serán presentados a través de un *dashboard*, el cual, de manera visual indicará las mediciones de CO₂ realizadas durante el día.

1.5. Estado del arte

Existen en el mercado sensores que permiten la medición de un solo parámetro y sensores de múltiples parámetros. La concentración de CO₂ a través de las partículas por millón existentes en el aire interior es una buena referencia para estimar la calidad del aire. Es posible encontrar partículas por millón en el ambiente en tres formas diferentes, incluidas partículas con diámetros inferiores a 10 μm (PM₁₀), partículas con un diámetro inferior a 2,5 μm (PM_{2,5}) y partículas con un diámetro inferior a 0,1 μm (PM_{0,1})[8]. Estudios revelan que las PM_{0,1} Y PM_{2,5} son las partículas con las cuales es posible medir la calidad que afecta la calidad del aire[9].

1.5.1. Sistemas embebidos existentes en el mercado, para el monitoreo de CO₂

Esta sección presenta un análisis breve de los dispositivos y las aplicaciones requeridas para el monitoreo de la calidad del aire en interiores a través de la conectividad *IoT*, El estudio que se realiza en en base a la medición de las partes por millón de CO₂ en el ambiente. El estado del arte proporciona un amplio panorama de los dispositivos ya existentes en este campo. La literatura muestra el gran potencial de la tecnología *IoT* para desarrollar soluciones inteligentes para mitigar los contagios generados por la pandemia del COVID-19.



Figura 1.2: Monitor de CO₂ comercial.

En el mercado encontramos una amplia variedad de sistemas de monitoreo de la calidad el aire, un ejemplo de dichos dispositivos es el mostrado en la figura 1.2.

Nuestra búsqueda se centrará en los dispositivos que tengan la capacidad de medir la concentración de CO₂ en espacios confinados y la detección de partículas PM_{0,1} y PM_{2,5} las cuales son las que afectan la calidad del aire.

Se presenta en la tabla 1.1 algunos dispositivos de diversas empresas las cuales ofrecen sistemas para el monitoreo de la calidad del aire.

Nombre del sistema	Parámetros de medición	Rango de medición del CO ₂	Tecnología empleada para el sensor CO ₂	Vida útil del dispositivo	Conectividad a Internet	Sistema de alarma o alerta
Sistema Kamysen	*CO ₂ *Humedad *Temperatura	400-5000 PPM	NDIR		No	Indicadores led
Sistema Langkou	*CO ₂ *Humedad *Temperatura	400-5000 PPM	NDIR	5 años	No	Alarma
Sistema M Abanopi	*CO ₂ *Humedad *Temperatura	0-9999 PPM	NDIR	15 años	No	Alarma
Sistema Gzair modelo 3	*CO ₂ *Humedad *Temperatura	0-9999 PPM	NDIR		No	
Air CO2ntrol Mini	*CO ₂ *Humedad *Temperatura	0-3000 PPM	Electroquímico		No	Indicadores led
Air CO2ntrol 3000	*CO ₂ *Humedad *Temperatura	0-3000 PPM	NDIR		No	Alarma sonora
Air CO2ntrol 5000	*CO ₂ *Humedad *Temperatura	0-5000 PPM			No	Alarma sonora
HOBO MX1102A	*CO ₂ *Humedad *Temperatura	0-5000 PPM			Sí	Alarma sonora
Qingping Monitor	*CO ₂ *Humedad *Temperatura	0-5000 PPM	NDIR	3 años	Sí	Alarma sonora

Tabla 1.1: Comparativa de los sistemas de monitoreo existentes en el mercado.

Se observa en la tabla 1.1 que la mayoría de los dispositivos comerciales, cuentan con un menor rango de trabajo para la detección de CO₂.

1.5.2. Comparativa con proyectos similares

Actualmente el desarrollo de sistemas embebidos con conectividad hacia la nube se encuentra en su apogeo. La importancia de sistemas robustos para facilitar la visualización de los datos ha sido una forma de estimar los riesgos de contagio de COVID-19, hoy en día los trabajos relacionados con este tema han crecido considerablemente para crear espacios de trabajo y convivencia óptimos en espacios cerrados.

A continuación, se muestra una revisión de trabajos similares encontrados en la literatura, se muestra una clara evidencia del mejoramiento de estos sistemas conforme se va conociendo el comportamiento del virus, finalmente se hace una comparativa de dichos sistemas con el trabajo descrito en esta tesis.

En el artículo [10], se aprecia un trabajo similar, haciendo uso de sensores electroquímicos para la detección de partículas de CO_2 en espacios cerrados, la metodología usada es similar, sin embargo cabe destacar que el uso de sensores de este tipo tienen una menor precisión y rango de medición, el sistema embebido planteado es dependiente de la conexión a internet, si la conexión falla el sistema no enviara las mediciones correspondientes hasta que exista una conexión a internet exitosa.

Es posible observar un proyecto similar en [11], se basa en una tecnología de microplacas calefactoras, donde algunos factores externos al sistema, como lo son la temperatura ambiental puede incrementar el error de medición, además de que el sensor hace mediciones de gases TVOC, no exclusivamente de CO_2 . De igual manera la dependencia de la conexión inalámbrica a un servidor local no permitiría hacer mediciones cuando no exista una conexión exitosa, las personas dentro del recinto no sabrán las condiciones de riesgo existentes al menos que vean la base de datos en la nube, ya que el sistema carece de notificaciones sonoras o visuales.

En el artículo [12], la implementación del sistema se realiza con varios kits, para experimentación y pruebas que son de gran ayuda, sin embargo para una implementación en campo no son recomendables ya que al tener ser un sistema modular las fallas de *hardware* pueden aumentar. Los sensores usados para la medición de CO_2 son de tecnología electroquímica, inferiores a los sensores NDIR implementados actualmente en la industria del monitoreo de la calidad del aire. Es importante

mencionar que dicho trabajo carece de una forma directa de notificación al usuario que se encuentra en el recinto a medir y la única forma de conocer las mediciones del sistema es contar con las credenciales del *software* IoT.

1.5.3. Conclusiones del estado del arte

Es importante puntualizar que varios sistemas de monitoreo basados en *IoT* carecen de la generación de notificaciones sonoras para alertar a los usuarios finales sobre los niveles óptimos de convivencia y sin riesgo de un contagio masivo en el lugar a monitorizar.

La gran mayoría de los dispositivos mostrados en el anterior análisis, no cuentan con sensores de alta precisión ni con un amplio periodo de trabajo, y necesitan de mantenimiento o calibración en un periodo largo de funcionamiento del sistema.

Los dispositivos que cuentan con conectividad a la nube tienen costos altos o incluso algunos es necesario adquirir algún complemento para la conexión a internet.

La mayoría de los sistemas desarrollados comercialmente y con bajos costos, no siguen los procedimientos de calibración adecuados ni las pruebas de confiabilidad antes del despliegue de los sensores IAQ (*Indoor air quality*) en las ubicaciones objetivo.

Ningún sistema comercial mostrado en la tabla 1.1 tiene una salida de activación para algún dispositivo de ventilación que ayude a mitigar los niveles altos de CO₂ en el ambiente, es necesario adquirir algún asistente virtual para la conexión de dispositivos inteligentes en recinto.

El sistema propuesto en este trabajo de tesis busca tener la conexión a internet del dispositivo con una base de datos el cual se almacene la información adquirida a través del sensor, esto será benéfico para trabajos en el futuro los cuales se haga una trazabilidad del comportamiento de los niveles de CO₂.

Es importante mencionar que nuestro sistema, si bien se basa en la tecnología *IoT*, se prevé en las fallas que pueden existir en caso de una conexión fallida y esta sea provocada por factores externos al sistema, de este modo el sistema planteado en esta tesis debe ser capaz de seguir funcionando y alertar de un posible riesgo de contagio a las personas

que se encuentren en el recinto, se ha planteado realizar estas alertas con notificaciones de voz, esto es un punto fuerte en nuestro trabajo.

Además, varios sistemas de monitoreo basados en *IoT* carecen de la generación de notificaciones o activadores para alertar a los usuarios finales sobre la disminución de las concentraciones de contaminantes en el medio ambiente. Se debe reducir el consumo de energía de los sistemas de monitoreo en tiempo real. La precisión de los sensores y el almacenamiento seguro de los datos debe ser otra preocupación importante para abordar los desafíos en el campo del *IoT*. [13].

2. Marco teórico

En este apartado se busca realizar un análisis brevemente de todas las bases teóricas para el desarrollo de esta tesis, se llevara acabo una breve descripción de las diversas tecnologías que actualmente se emplean para la monitorización de la calidad del aire relacionadas con la conexión de sistemas embebidos hacia internet.

En los últimos años, la calidad del aire en interiores (*IAQ*) ha recibido una atención considerable de gobiernos ambientales, instituciones políticas y la comunidad de científicos internacionales, debido a su estrecha asociación con la salud pública, la comodidad y el bienestar de las personas[14][15].

El *IoT* es un tema ampliamente discutido en el campo de las tecnologías de la comunicación. Esta tecnología puede conectar una cantidad ilimitada de dispositivos y sensores para influir en la forma en que trabajamos y vivimos. *IoT* encuentra múltiples aplicaciones en escenarios del mundo real mientras automatiza procesos para aumentar la productividad[15]. La información registrada puede ser vista por expertos en salud pública, funcionarios gubernamentales y legisladores. El monitoreo ambiental es una aplicación crítica de *IoT* y ayuda a mejorar la salud y el bienestar público[16].

Actualmente se puede hacer uso de las tecnologías emergentes para advertir sobre las condiciones de riesgo de contagio del COVID-19 como un auxiliar para mitigación del virus en espacios cerrados, como se muestra en la figura 2.1, el Internet de las cosas (*IoT*) muestra un gran potencial para medir datos en tiempo real de un ambiente cerrado que pueden ayudar aún más a los ocupantes del edificio a tomar decisiones relevantes para tomar medidas necesarias de ventilación y así mitigar los niveles de CO₂.

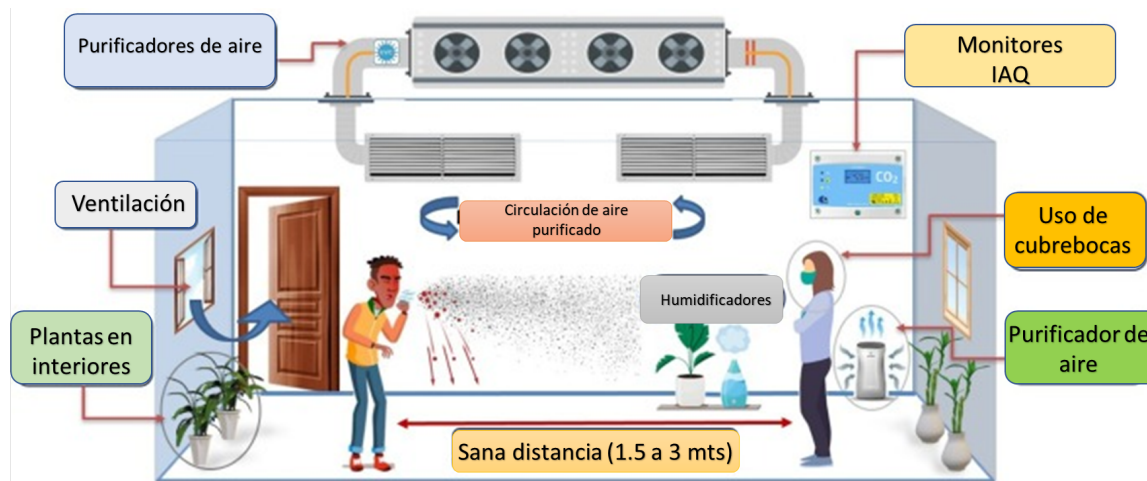


Figura 2.1: Técnicas de mejora para la calidad del aire interior (IAQ) para el combate del COVID-19.

Sin embargo, la comunidad científica enfrenta varios desafíos al seleccionar sensores para diseñar sistemas de monitoreo en tiempo real. Es fundamental hacer un análisis cualitativo y cuantitativo para la evaluación de sensores mientras se diseñan sistemas de monitoreo eficientes. Este capítulo proporciona puntos importantes donde se analizan aspectos relevantes para la conexión del monitoreo de CO_2 y la conectividad del *IoT* para el procesamiento de los datos adquiridos [13].

Existen principalmente dos métodos potenciales que respaldan el desarrollo de sistemas de monitoreo de la calidad del aire en espacios residenciales y comerciales: el *IoT* y las redes de sensores inalámbricos[17]. Este trabajo se centra en el desarrollo de los sistemas *IoT* ya que se ha demostrado su ventaja con la capacidad de manejar y transferir datos masivos a través de internet, es por esta razón que este tipo de dispositivos están ganando más popularidad para las aplicaciones de monitoreo de variables en edificio inteligentes. Las tecnologías de comunicación y las soluciones de *Hardware* respaldan el monitoreo confiable de la calidad del aire para mejorar la salud ambiental y el bienestar público [18]. Los sistemas de monitoreo incluyen un conjunto de sensores para medir los parámetros deseados, un microcontrolador para recopilar datos de las unidades de sensores, puertas de enlace y sistemas de comunicación. La arquitectura general del sistema de monitoreo de *IoT* se muestra en la figura 2.2. Los datos obtenidos de los sistemas de monitoreo se pueden almacenar en la nube para facilitar el acceso en el futuro. Puede ayudar a

los ocupantes del edificio a tomar una decisión relevante sobre la mejora de la calidad del aire en las instalaciones del edificio.

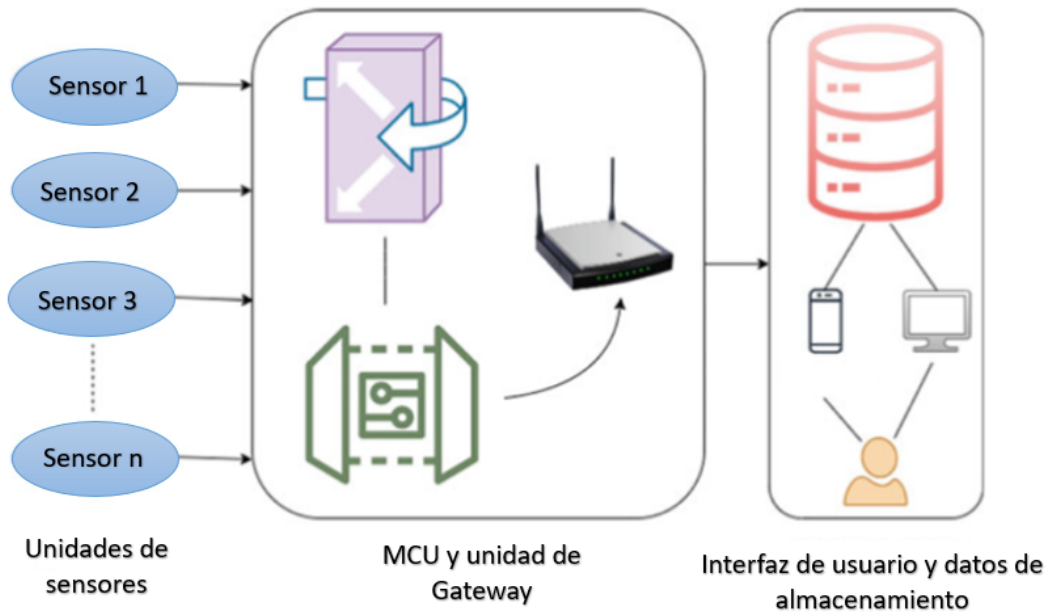


Figura 2.2: Diagrama general de los sistemas *IoT*.

A continuación se muestra en la tabla 2.1, las herramientas de *Hardware* y *Software* más usadas actualmente en la elaboración de trabajos de *IoT*, posteriormente se hará un análisis de las herramientas seleccionadas y usadas en este trabajo de tesis.






SoCs	Módulos	Tarjetas de desarrollo	Firmware, SDK e IDE	Plataformas de servicio en la nube
 <ul style="list-style-type: none"> -ESP32-S -ESP32-S2 -ESP8266 -ESP8285 -ESP32-C2 -ESP32-C3 -ESP32-C6 	 <ul style="list-style-type: none"> -ESP32- WROOM-32 -ESP32-S3-WROOM-1 -ESP32-S3-WROOM-1U -ESP32-S3-MINI-1 -ESP32-S3-1U 	 <ul style="list-style-type: none"> -ESP32-S3-EYE -ESP32-S3-BOX -ESP32-WROOM-32 DEVKITV1 -ESP32-WROVER-B -ESP32-WRROM-32U 	 <ul style="list-style-type: none"> -Microphyton -CircuitPhyton -Arduino -ESP-IDF -Platformio -Eclipse -Visual Studio 	 <ul style="list-style-type: none"> -Adafruit IO -Grafana -Ubidots -Thingspeak -Azure -Aws -Google Cloud -Salesforce -Node Red

Tabla 2.1: *Hardware y software* comúnmente usado en la elaboración de trabajos de *IoT*.

2.1. Sistema embebido

Los sistemas embebidos son sistemas de cómputo con una integración de *hardware y software*, que son diseñados para realizar una función específica, se observa en la figura 2.3 los componentes más usuales de un sistema embebido. El término embebido o empotrado hace referencia al hecho que la electrónica o el sistema electrónico de control es una parte fundamental del sistema en que se encuentra[19].

Las características de un sistema embebido son las siguientes[20]:

- Tienen un procesador central: FPGAS, microprocesadores, micro-controladores, SoC, DSP, o incluso una mezcla de los anteriores.
- Son sistemas confiables
- Requieren un nulo o poco mantenimiento.
- Son sistemas seguros, tienen protección del *software* incrustado.
- Son eficientes en cuanto al consumo de energía.
- Son de propósito específico.

- Cuentan con una interfaz de usuario simple o carecen de ella, Algunos de ellos cuentan con interfaz gráfica de usuario lo cual facilita su programación, configuración y/o control. Cabe resaltar que su interfaz no es de uso general, debido a que es de uso específico.

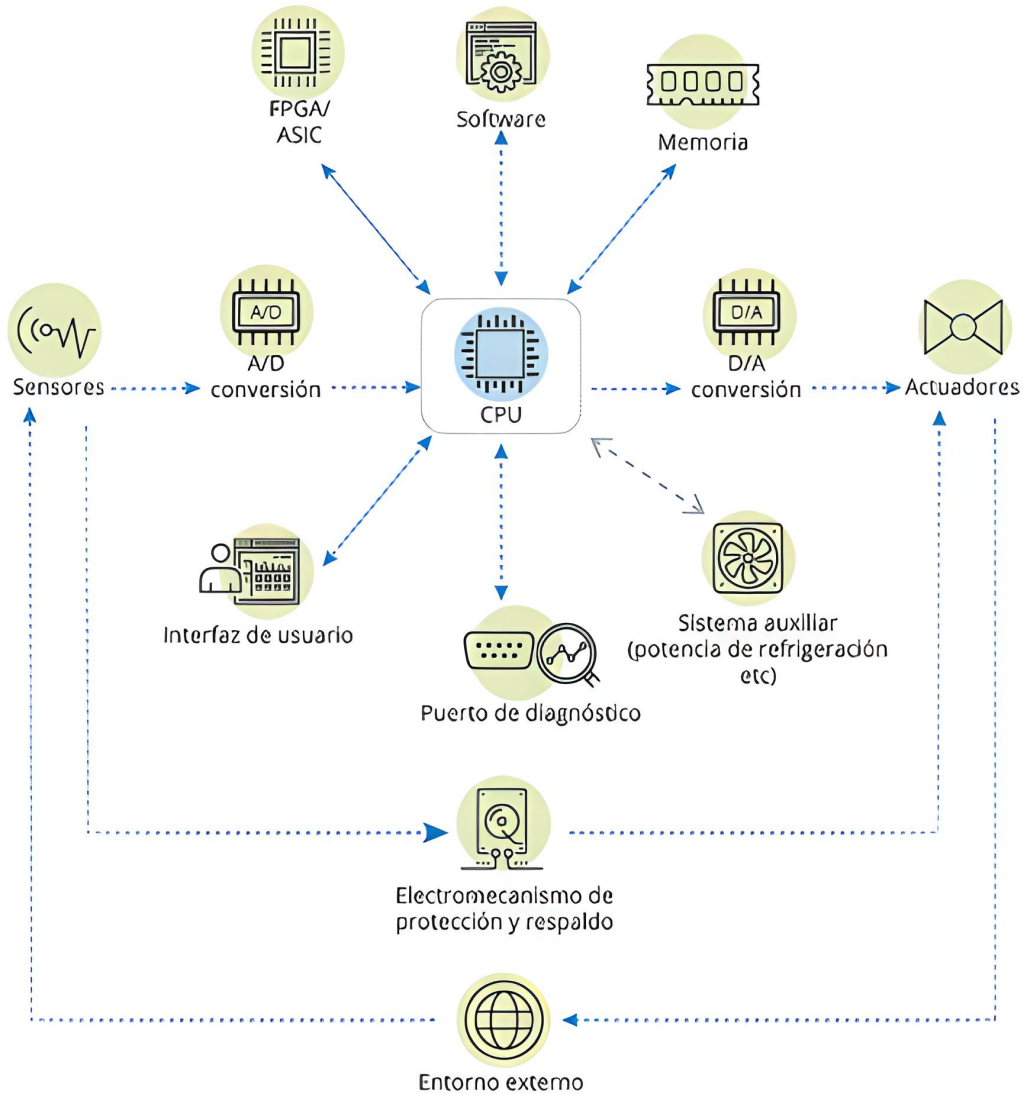


Figura 2.3: Componentes de los sistemas embebidos.

2.1.1. Diseño de placas de circuito impreso (PCB) para sistemas embebidos

Una placa de circuito impreso (PCB) es una placa hecha para conectar componentes electrónicos entre sí. Estos se utilizan en casi todas las computadoras y la electrónica de hoy. La tarjeta está hecha de un material que no conduce la electricidad, generalmente fibra de vidrio. Por

lo general, el cobre está grabado (fijado en líneas finas llamadas pistas) dentro del tablero entre las capas de fibra de vidrio o en la superficie del tablero. Esto hace que la electricidad fluya a través de las pistas originadas con cobre. Posteriormente los componentes electrónicos se unen a esta placa utilizando un metal para conducir la electricidad. El metal grabado en la placa permite que la electricidad viaje de un componente a otro en los circuitos eléctricos [21].

Los sistemas embebidos para cumplir con un óptimo funcionamiento son necesarios diseñarlos en placas de circuito impreso, por lo cual tenemos diversas opciones de *software* en mercado, las más utilizados se presentan en la figura 2.4.



Figura 2.4: *Software* comúnmente usado en el diseño de placas electrónicas.

2.1.2. *Software* de desarrollo para el diseño de PCB: *Eagle*

Se ha optado por hacer uso de la herramienta de diseño de la empresa *Autodesk* denominado *Eagle* ya que es un *software* de automatización de diseño electrónico (EDA) que permite a los diseñadores de placas de circuito impreso conectar sin problemas diagramas esquemáticos, ubicación de componentes, enrutamiento de PCB, contiene una biblioteca extensa de componentes electrónicos [22].

Ya que la empresa *Autodesk* tiene una amplia gama de *software* para desarrollo de ingeniería de diversas áreas, es posible hacer uso de la he-

ramienta *Fusion 360*, de esta forma unificar el diseño ECAD (*Electronic Computer-Aided Design*) donde la ingeniería, la electrónica y la manufactura se unifican con *Fusion 360* haciendo uso de la PCB desarrollada en *Eagle* como se muestra en la figura 2.5.

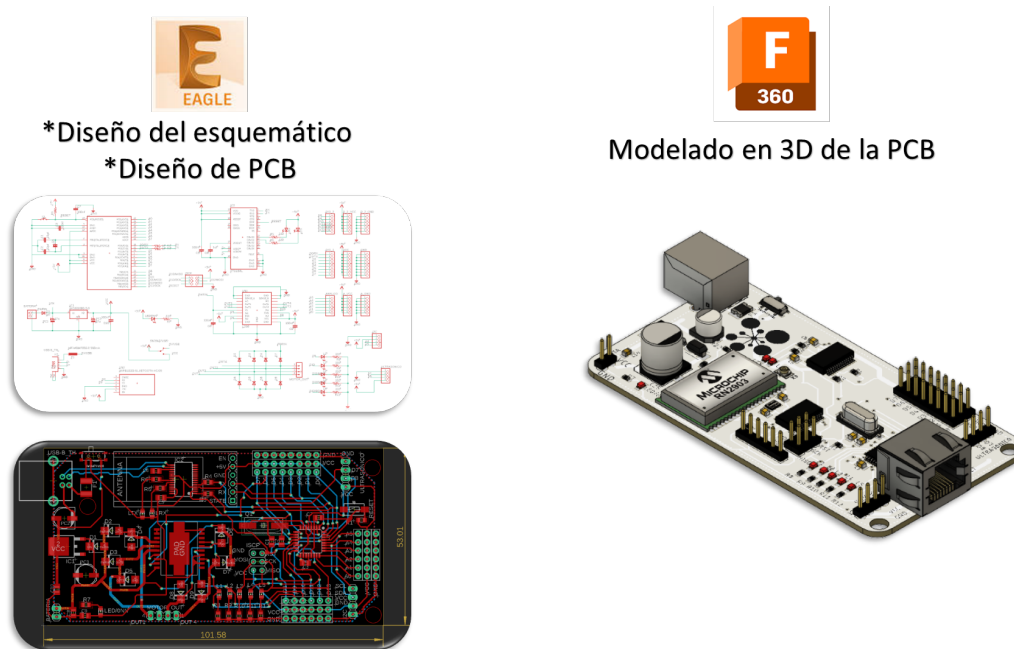


Figura 2.5: Integración de los *software Eagle* y *Fusion 360*.

2.2. Sensores para el monitoreo de la calidad del aire y sus parámetros

La monitorización de contaminantes en ambientes interiores requiere una adecuada selección de sensores. Las instalaciones del edificio pueden tener una variedad de contaminantes y las concentraciones pueden cruzar niveles de umbral críticos dependiendo de las actividades humanas, los materiales de construcción, los arreglos de calefacción e incluso los contaminantes generados por cocinas o espacios de preparación de alimentos[23].

La calidad del aire en interiores se ve afectada por varios parámetros de confort térmico especialmente por la temperatura y la humedad[24]. Existen otros parámetros como lo son PM_{10} , $PM_{2,5}$, CO_2 , CO , VOC , y NO_x [25].

Sin embargo, este trabajo se centra en solo aquellos sensores con

la capacidad de medir el CO_2 . Para dicho parámetro se ha utilizado una variedad de sensores que incluyen sensores electroquímicos, láser, analógicos, digitales entre otros[26].

La exposición a partículas finas ($\text{PM}_{2,5}$), ozono y otros componentes del aire contaminado provocan procesos de estrés oxidante e inflamación de las vías respiratorias y los pulmones ocasionando efectos adversos a la salud de las personas en el corto y largo plazo y alteran de manera importante la respuesta del sistema inmunológico.

En aproximadamente un 15 % de los pacientes, el COVID-19 afecta a los pulmones produciendo una neumonía que puede progresar rápidamente y comprometer la vida del paciente[27].

Para medir los contaminantes potenciales del aire en espacios cerrados, la literatura proporciona algunos ejemplos de los sensores más usados en sistemas embebidos y muestran evidencia de los resultados obtenidos al hacer uso de éstos dispositivos para la monitorización de la calidad del aire, existen sensores de un sólo parámetro y algunos otros que se centran en múltiples variables[28].

La mayoría de los investigadores recomiendan los sensores de la serie MQ para realizar mediciones de gases en el ambiente interior [28].

Los investigadores sugieren los sensores de la serie MiSC (6841 y 4514) o incluso la serie Figaro (TGS2610 y TGS2620) para la medición de múltiples gases que afectan la calidad del aire en ambientes cerrados [29]. El sensor *DFRobot Gravity BME680* es ampliamente recomendado para medir VOC (Compuestos orgánicos volátiles). La principal ventaja de usar este sensor es que puede eliminar la necesidad de invertir en sensores de confort térmico adicionales, ya que es capaz de medir la presión atmosférica, la humedad y la temperatura [30].

Hay varios tipos de monitores de CO_2 en el mercado, una buena opción para el sensado en estos dispositivos es hacer uso de sensores infrarrojos no dispersivos (NDIR). Estos sensores estiman la cantidad de CO_2 presente midiendo la cantidad de luz que se absorbe en longitudes de onda específicas, debido a lo cual la interferencia de otros gases presentes es mínima, aunque la humedad y la temperatura pueden afectar la lectura[31]. Numerosos sensores NDIR- CO_2 están disponibles, la precisión varía ampliamente y el precio no siempre es un indicador de calidad.

Un ejemplo de estos sensores puede ser el módulo de gas infrarrojo MH-Z19B mostrado en la figura 2.6, es un dispositivo desarrollado por la estrecha integración de tecnología madura de detección de gas absorbente infrarrojo, diseño de circuito óptico de precisión y un amplio rango de detección[32].



Figura 2.6: Sensor NDIR Mhz19-B.

Otro buen ejemplo de la aplicación de la tecnología NDIR es el sensor *SenseAir s8* diseñado para una fácil integración en sistemas embebidos, tiene un amplio rango de detección y no requiere mantenimiento a corto plazo[33].

Para la selección del sensor es importante analizar el rendimiento general del sistema y la tasa de error de las mediciones realizadas a través del sensor, esto con el fin de mantener un rendimiento confiable, ya que es importante trabajar en los requisitos de calibración y la frecuencia de mantenimiento.

2.2.1. Principio operativo de los sensores infrarrojos NDIR

La tecnología NDIR es un tipo de espectroscopia IR, se basa en el principio de que las moléculas de gas absorben la luz IR y la absorción de un determinado gas se produce a una longitud de onda específica. Por lo general, se usa una termopila con un filtro incorporado para detectar la cantidad de un gas específico. Por ejemplo, dado que el CO_2 tiene una fuerte absorbancia a una longitud de onda de $4,26 \mu\text{m}$, se utiliza un filtro de paso de banda para eliminar toda la luz fuera de esta longitud de onda. La figura 2.7 muestra el principio básico de un sensor de gas empleando la tecnología NDIR .

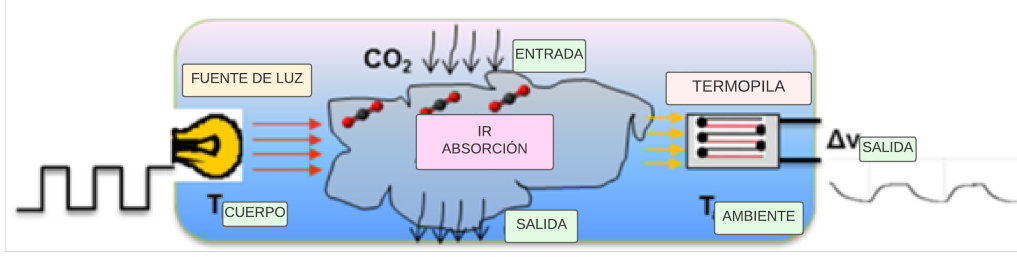


Figura 2.7: Principio del sensor de gas NDIR.

Las moléculas de gas absorberán la energía de radiación de la emisión de la lámpara. La absorción sigue la ley de Lambert-Beer:

$$I = I_o * e^{-kcl} \quad (2.1)$$

Donde I es la intensidad de IR transmitida en el lado del detector de la termopila, I_o es la intensidad inicial en la fuente de IR, k es el coeficiente de absorción específico del gas objetivo, c es la concentración de gas y l es la longitud de la ruta de absorción de la fuente de luz al detector de termopila. La termopila se utiliza para detectar el cambio de intensidad de la luz. Su voltaje de salida será el siguiente:

$$V = n * \Delta\alpha * (T_{Cuerpo} - T_{Ambiente}) \quad (2.2)$$

Donde $\Delta\alpha$ es la diferencia de los coeficientes de Seebeck de los materiales de la termopila y n es el número de termopares en detector de termopila. T_{Cuerpo} es la temperatura del cuerpo negro que emite radiación térmica (es decir, la lámpara IR), y $T_{Ambiente}$ es la temperatura del ambiente circundante.

Dentro de la cámara de gas, la energía de radiación de la lámpara IR podría considerarse como radiación de cuerpo negro ideal. La radiación emitida por un cuerpo negro como resultado de la diferencia de temperatura entre el cuerpo negro y el ambiente se conoce como radiación térmica. De acuerdo con la ley de Stefan-Boltzmann, la radiación térmica por unidad de área se expresa con la siguiente ecuación:

$$R_T = \sigma * (T_{Cuerpo}^4 - T_{Ambiente}^4) \quad (2.3)$$

$$R_T = \sigma * (T_{Cuerpo}^4 - T_{Ambiente}^4) \quad (2.4)$$

Asumiendo que no hay pérdida en la intensidad de la luz mientras viaja a través de la cámara, entonces $R_T = I$. Después de reorganizar las ecuaciones anteriores, la ecuación para el voltaje de salida de la termopila se convierte en:

$$V = \frac{n * \Delta\alpha * [I_o * e^{-kcl}]}{\sigma * (T_{Cuerpo}^2 + T_{Ambiente}^2) * (T_{Cuerpo} + T_{Ambiente})} \quad (2.5)$$

Examinando la ecuación 2.5 se aprecia que el voltaje de salida de la termopila se verá afectado por la temperatura ambiente y la incertidumbre de la intensidad de la lámpara IR con una relación compleja. Para mantener una mejor precisión del sistema, se debe tener especial consideración en el diseño implementado. Podemos ver que la compensación de temperatura es una forma efectiva de mantener la precisión del sistema. Para lograr esto, los termistores se integran comúnmente en el sensor de termopila y su resistencia cambia según la temperatura ambiente circundante. Para una mejor precisión en la medición, tener un voltaje constante estable para excitar el termistor es una buena opción.[31]

2.2.2. Sensor *SenseaAir S8*

El sensor de detección de partículas de CO_2 es parte fundamental de este trabajo de tesis, ya que se busca una mejora en la etapa de sensado en comparación de tecnologías empleadas por la mayoría de monitores de CO_2 que emplean frecuentemente tecnologías como la oxidación catalítica.

Anteriormente se ha descrito la cualidad de la tecnología de sensores infrarrojos NDIR, los cuales actualmente han ganado un gran auge en la implementación de monitores de CO_2 .

Es por ello que se el *SenseaAir S8* (Figura 2.8) es una elección para el uso de sistemas embebidos orientados a la detección de partículas perjudiciales para el ser humano. A continuación se muestra una breve

descripción del funcionamiento, así como una de las principales características del mismo y finalmente se hace un análisis de las razones por cuáles se realizó la elección de dicho sensor.



Figura 2.8: *SenseAir S8*.

Este sensor es la opción ideal para el control de la ventilación interior y el monitoreo de CO₂ en aplicaciones residenciales. El sensor se tiene cómo principio de funcionamiento la moderna tecnología de infrarrojos (NDIR). El *SenseAir S8 Residential* está diseñado para una fácil integración en sistemas embebidos [34].

Se muestra en la tabla 2.2 algunas de las características más notables del sensor.

Parámetros	Especificaciones SenseAir S8
Gas objetivo	Dióxido de carbono (CO ₂)
Principio de operación	NDIR (Infrarrojo No Dispersivo)
Rango de medición	400 a 2,000 ppm (0 a 10,000 ppm en modo extendido)
Temperatura de operación	0 a 50° C
Tiempo de respuesta	2 minutos para el 90%
Voltaje de alimentación	4.5 v. a 5.25 v. DC
Comunicación empleada	UART (Modbus)
Dimensiones (Ancho, largo, alto)	8.5 x 33.5 x 20 mm
Corriente de consumo	300 mA pico, 30 mA promedio
Esperanza de vida	15 años

Tabla 2.2: Especificaciones más notables del *SenseAir S8*.

El *SenseaAir S8* es distribuido por la empresa sueca SenseAir, un proveedor global líder de tecnología de detección de aire y gas, actualmente este sensor no cuenta con falsificaciones cómo lo son el caso de otros sensores cómo el popular MHZ19-B o el MHZ-19C, teniendo un precio relativamente parecido a los sensores antes mencionados.

El *SenseaAir S8* tiene un sistema de calibración automática (ABC, por sus siglas en inglés *Automatic Baseline Correction*) muy bueno por lo que, en condiciones de espacios cerrados en interiores, no habrá una necesidad de calibración manual. El algoritmo ABC vigila permanentemente la lectura de CO₂ mínima durante un periodo preconfigurado y corrige lentamente cualquier deriva en el largo plazo comparado con la concentración de 400 ppm de CO₂ óptimo para ambientes interiores. [34]. Sin embargo cuando se utiliza el sensor en modo extendido las mediciones de CO₂ < 400 ppm no resultan precisas, además el algoritmo ABC para aplicaciones industriales , el sensor requiere mantenimientos más habituales, en comparación de aplicaciones comunes cómo en interiores.

Este sensor contiene una salida de colector abierto para la conexión de un indicador de alarma, con un voltaje de salida de 0.3v a 5.5v con una corriente máxima de 100 mA.

La conexión con microcontroladores del *SenseaAir S8* para su comunicación habitualmente es por el protocolo UART, sin embargo también cuenta con una salida PWM para proporcionar una salida analógica.

Existen cuatro versiones diferentes del *SenseaAir S8*, a continuación se presenta en la tabla 2.3 la descripción de dichas variantes y las principales funcionalidades.

Matricula	Sensor	Descripción
004-0-0010	S8 Commercial	Conexión con conectores macho (para control de ventilación y monitoreo de CO ₂)
004-0-0013	S8 Residencial	Para conexión sin conectores macho (para control de ventilación y monitoreo de CO ₂)
004-0-0056	S8 Residencial	Conexión con conectores macho (para control de ventilación y monitoreo de CO ₂)
004-0-0017	S8 5%	Conexión con conectores macho (para aplicaciones de control y alarma con margen de error de precisión del $\pm 5\%$)
004-0-0050	S8 2%	Conexión con conectores macho (para aplicaciones de control y alarma con margen de error de precisión del $\pm 2\%$)
004-0-0053	S8 LP	Para aplicaciones en las que tanto el consumo de energía como la precisión son factores críticos

Tabla 2.3: Variaciones del *SenseAir S8*.

Se ha seleccionado el sensor *SenseAir S8 LP* 004-0-0053 por su alta precisión y bajo consumo de energía, el proveedor asegura la calibración individual de cada sensor, además cuenta con una interfaz digital UART[34], lo cuál será de fácil implementación para comunicarse con el SoC a usar. El sensor no requiere mantenimiento en condiciones óptimas de ambientes cerrados y tiene una vida útil estimada de más de 15 años, sus cualidad son idóneas para el proyecto sumando su bajo precio y sus altas prestaciones.

2.3. Microcontroladores y SoC

El dispositivo el cual se encargará de hacer el procesamiento de la información generada por el sensor NDIR, se puede hacer con MCU (microcontrolador) o SoC (sistema en un chip), es por esta razón que requiere la selección de dispositivos confiables en función de los requerimientos del proyecto.

Se pueden encontrar diversos trabajos científicos y académicos donde la Raspberry Pi y Arduino han sido la opción preferida para diseñar MCU de puerta de enlace. Existen opciones tales como el módulo

ESP8266 para el desarrollo de la puerta de enlace. Estos dispositivos SoC son muy usados ya que sus prestaciones y los componentes integrados en las tarjetas de desarrollo son muy fáciles de adquirir a bajos costos [35].

Sin embargo, actualmente el SoC de ESP32 mostrado en la figura 2.9 es una excelente opción para el desarrollo de sistemas embebidos con conectividad *IoT* ya que el soporte de la comunidad científica y académica se ha centrado en este dispositivo que se dirige a una amplia variedad de aplicaciones, que van desde redes de sensores de bajo consumo hasta las tareas más exigentes, como la codificación de voz, la transmisión de música y la decodificación de MP3, además de contar con conectividad inalámbrica *Bluetooth* y *Wi-Fi* en un solo módulo. De igual manera el costo de estos dispositivos es accesible y sus funcionalidades son amplias para el desarrollo de proyectos de *IoT* [36].

Más adelante se analizará con mayor profundidad las prestaciones y características del ESP-WROOM 32 ya que este dispositivo será el que se utilizará para el desarrollo del sistema embebido de esta tesis, se hará una análisis y una justificación de su uso.



Figura 2.9: Módulo ESP 32.

Existen múltiples versiones de Arduinos disponibles en el mercado y cada una de ellas viene con características, especificaciones y precios únicos. Arduino Uno es la placa de desarrollo generalmente más utilizada la cual se muestra en la figura 2.10, cuenta con microcontrolador Atmega 328P con soporte extendido a 32 KB de memoria de programa, 2 KB de RAM, 1 KB de EEPROM. Este microcontrolador tiene 14 pines de entrada y salida digital junto con 6 entradas analógicas, es compatible con un voltaje de alimentación de 5 volts[37]. Una opción

ampliamente usada por su versatilidad son el Arduino Nano Y es una versión más pequeña de Arduino Uno que es adecuada para la implementación en espacios reducidos. Igual que Arduino Uno, también funciona con un microcontrolador Atmega 328P [38]. Sin embargo estos dispositivos antes mencionados, no cuenta con la conectividad a internet y no son comúnmente usados para aplicaciones de monitoreo inteligente en tiempo real ya que es necesario agregar un módulo el cual proporcione este requisito, la ventaja es la rentabilidad y un gran soporte por parte de la comunidad científica y aficionados usando estos dispositivos.

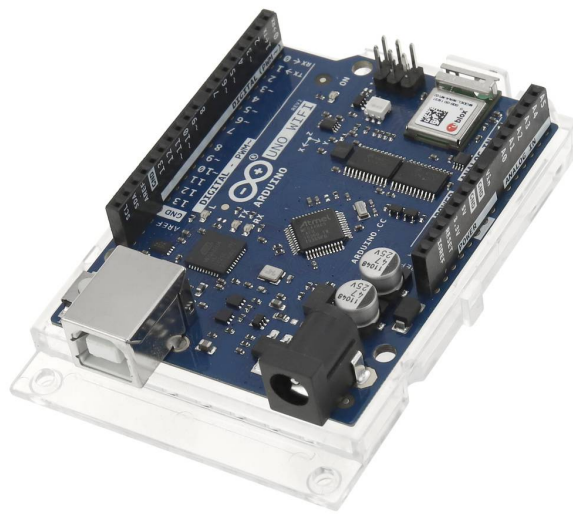


Figura 2.10: Arduino Uno *WiFi*.

Por otro lado la empresa Arduino proporciona opciones orientadas al *IoT* básico y con el factor de forma estándar de las placas de desarrollo de la familia Arduino Uno. Un ejemplo es la placa Arduino Uno *Wi-Fi* Rev. 2, la cual se observa en la figura 2.10, que es una excelente opción para la creación de redes de sensores conectadas a un punto de acceso a internet, incluso se cuenta con la funcionalidad de *Bluetooth* de baja energía, es una solución integral para muchos de los escenarios básicos de aplicaciones de *IoT* [39].

No obstante a pesar de ser un dispositivo con características importantes para el desarrollo de aplicaciones *IoT*, la familia de Arduino *Wi-Fi* Uno, no es de fácil acceso y sus precios se elevan considerablemente en comparación con otras tarjetas de desarrollo.

Los SoC basados en Raspberry son una opción ampliamente usadas

en aplicaciones *IoT*. El módulo básico Raspberry Pi Zero generalmente viene con 512 MB de RAM y ARM11 de un solo núcleo de 1 GHz. El Raspberry Pi 2B está disponible con 1 GB de RAM y es compatible con Quad-Core ARM Cortex-A7 de 900 MHz. La Raspberry Pi 3B es una versión un poco más alta con 1 GB de RAM y un procesador ARM Cortex A53 de cuatro núcleos y 64 bits a 1,2 GHz. Los investigadores y desarrolladores de sistemas embebidos también han usado Raspberry Pi3B+ en el pasado para diseñar sistemas de monitoreo de IAQ (*Indoor air quality*). Actualmente la Raspberry Pi 4 la cual es mostrada en la figura 2.11 es una opción de uso para el monitoreo de la calidad del aire, sus prestaciones con procesador a un ARM Cortex-172 con cuatro núcleos a 1,5 GHz también implicaba pasar de los 40 nm a los 28 nm. En consecuencia, todos los componentes y la potencia del dispositivo ha cambiado en comparación de la versión anterior, el amplio número de pines para la conexión de dispositivos entrada- salida y su amplia capacidad de procesamiento de información hacen este dispositivo una excelente opción para aplicaciones *IoT* [40].



Figura 2.11: Raspberry pi 4 Modelo B.

Otras opciones disponibles para que las MCU diseñen sistemas inteligentes de medición de contaminantes basados en *IoT* son el módulo BeagleBone, Texas Instruments CC3200, STM32F103C8T6(ARM), Placa Intel Edison, ARM Cortex-M0 y el módulo MSP430F5529, sin embargo, la gran mayoría de estos sistemas no tienen un costo accesible.

2.3.1. ESP-32

La aplicación del SoC ESP-32 en esta tesis se usará a través del módulo ESP-WROOM-32, según la clasificación de la empresa Espressif Systems[41], el SoC se refiere a un chip que integra todas o la mayor parte de componentes necesarios para el funcionamiento de un sistema electrónico. Entre ellas se incluye una CPU, por lo que podemos hablar de un procesador que incluye más componentes en su interior que normalmente estarían relegados a chips externos a este.

Éstos SoCs se pueden diferenciar por la presencia de uno o dos núcleos, si cuentan con memoria flash, el tipo de conexión *Wi-Fi* y el tamaño del encapsulado, la mayoría de los SoCs de la empresa *Espressif Systems* los encontramos en un encapsulado de montaje superficial tipo QFN.

Se define como módulo al sistema contenido en un encapsulado metálico, el cual contiene en su interior el sistema mínimo para el uso del SoC, la principal característica de los módulos es la fácil integración a una PCB y a otros componentes externos al sistema mínimo.

Finalmente se define como tarjetas de desarrollo a los circuitos electrónicos que contienen al módulo y otros elementos como puertos, conectores y reguladores que les permita a los usuarios acceder fácil y rápidamente a los periféricos para realizar aplicaciones y pruebas.

Se puede observar de manera gráfica las secciones del ESP-32 a partir de una tarjeta de desarrollo en la figura 2.12.

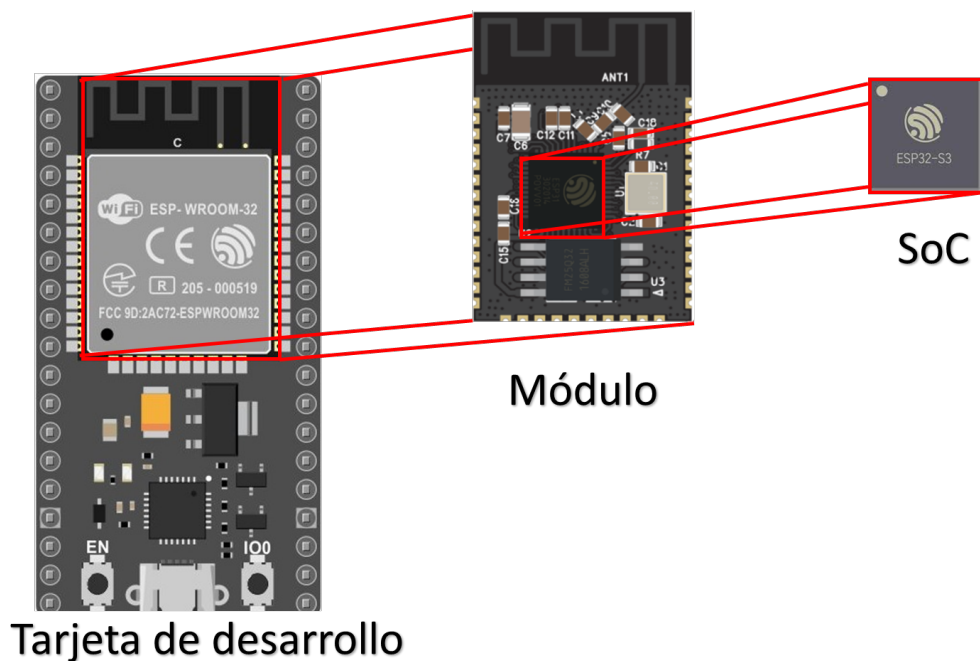


Figura 2.12: Esp-32 seccionado.

El desarrollo de la etapa de control electrónico de esta tesis será con el módulo ESP32-WROOM-32, a continuación se muestran las especificaciones técnicas de dicho módulo en la tabla 2.4:

Especificaciones técnicas	Descripción
Voltaje de Alimentación	3.3 VDC (2.7~ 3.6V)
SoC	ESP32 (ESP32-D0WDQ6)
Voltaje lógico entradas/salidas	3.3 VDC.
CPU	Dual Core <i>Tensilica Xtensa LX6</i> (32 bit)
Frecuencia de Reloj	240 MHz.
SRAM	520 KB.
Memoria Flash Externa	4MB (Contenido dentro del módulo)
Pines GPIO	34 (Incluyendo todos los periféricos)
UART	3
SPI	3
I ² C	2
PWM	16 canales independientes (16-bits)
ADC	12 (12-bit)
DAC	2 (8-bit)
Timers	3 (16-bit)
Conectividad <i>Wi-Fi</i> :	802.11 b/g/n/e/i (802.11n a 2.4 GHz hasta 150 Mbit/s)
<i>Bluetooth</i> :	v4.2 BR/EDR and <i>Bluetooth Low Energy (BLE)</i>
Antena	Montada en PCB
Protocolos de redes	IPv4, IPv6, SSL, TCP/UDP/HTTP/FTP/MQTT
Rango de frecuencia del <i>Wi-Fi</i>	Frecuencia: 2.4 2.5 GHz.
Dimensiones físicas	18*25.5*3.1 mm.
Peso	3 gr.

Tabla 2.4: Especificaciones técnicas ESP32 WROOM.

El ESP-WROOM-32 es un potente módulo que integra *Wi-Fi* y *Bluetooth* de bajo consumo, ideal para desarrollar productos de *IoT*. La integración de *Bluetooth* LE y *Wi-Fi* permite una amplia gama de aplicaciones, el uso de *Wi-Fi* permite una comunicación de conexión a red LAN, mientras que el *Bluetooth* nos permite conectarse directamente a otro dispositivo como un *smartphone*.

En el núcleo de este módulo se encuentra el SoC ESP32-D0WDQ6. El chip integrado está diseñado para ser escalable y adaptado. Hay dos núcleos de CPU que se pueden controlar individualmente, y la frecuencia del reloj es ajustable de 80 MHz a 240 MHz.

ESP32 integra un amplio conjunto de periféricos como sensores táctiles capacitivos, sensores Hall, amplificadores de bajo nivel de ruido, interfaz para SD, *Ethernet*, SPI, UART e I²C.

El SoC ESP32 de Espressif Systems es la evolución del ESP8266, diseñado para superar a su antecesor en capacidad de procesamiento y conectividad, integra un potente microcontrolador con arquitectura de 32 bits, conectividad *Wi-Fi* y *Bluetooth*. El SoM(*System on Module*) ESP-WROOM-32 fabricado por Espressif integra en un módulo el SoC ESP32, memoria FLASH, cristal oscilador y antena *Wi-Fi* en PCB[34].

El módulo ESP-WROOM-32 permite el desarrollo de aplicaciones en diferentes lenguajes de programación, frameworks y bibliotecas de fácil acceso. Los más comunes a elegir son: Arduino, Esp-idf(*Espressif IoT Development Framework*) desarrollado por el fabricante del chip, *Platformio*, *MicroPython*, *CircuitPython*, *LUA*, *Javascript* (*Espruino*, *Duktape*, *Mongoose JS*), inclusive cuenta con un sistema operativo precargado cómo firmware, sin embargo también es posible usar RTOS's (Sistemas operativos de tiempo real) como lo son: *Zephyr Project*, *Mongoose OS*, *NuttX RTOS*.

Una solución única y novedosa que está ganando una amplia aceptación e impulso es acoplar el microcontroladores y SoCs de bajo nivel del *hardware* con un lenguaje de programación de alto nivel como lo es *Python*. Una de esas soluciones es *MicroPython*. Este se ejecuta en diferentes piezas de proveedores de microcontroladores y es de código abierto, por lo que está fácilmente disponible para el uso personalización de los desarrolladores según sus propias necesidades. A continuación se podrá conocer un poco más de dicho lenguaje de programación él cual

es el elegido para hacer uso en esta tesis.

2.4. *Python* para microcontroladores: *MicroPython*

Los sistemas electrónicos embebidos requieren un profundo conocimiento en arquitectura de microcontroladores, sensores, algoritmos, protocolos de internet y diversas aplicaciones de usuario final. Con tiempos de desarrollo más cortos y con más exigencias en las características de estos sistemas, los equipos de desarrollo necesitan encontrar formas de acelerar ciertos procesos, como lo son el diseño y la programación.

Existen en el mercado diversas plataformas microcontroladores que están disponibles para ayudar a acelerar el proceso de desarrollo, pero el problema con estas soluciones es que limitan los desarrolladores a un único proveedor de microcontroladores y es difícil migrar a otros dispositivos, si esto fuese necesario.

Una solución ante dicha problemática y que actualmente está ganando una amplia aceptación ante la comunidad de ingenieros y científicos es acoplar el microcontrolador de bajo nivel del *hardware* con un lenguaje de programación de alto nivel como lo es *Python*. Una de esas soluciones es *MicroPython*. Este se ejecuta en diferentes piezas de proveedores de microcontroladores y es de código abierto, por lo que está fácilmente disponible para el uso personalización de los desarrolladores según sus propias necesidades[42].

En *MicroPython* podemos encontrar una implementación sencilla y eficiente del lenguaje de programación *Python 3* que incluye un pequeño subconjunto de la biblioteca estándar de *Python* y está optimizado para ejecutarse en microcontroladores y en entornos restringidos.

MicroPython está repleto de funciones avanzadas, como un indicador interactivo, enteros de precisión arbitraria, cierres, comprensión de listas, generadores, manejo de excepciones y más. Sin embargo, es lo suficientemente compacto para ser almacenado y funcionar con solo 256 Kbytes de espacio en memoria flash y 16 Kbytes de memoria RAM, sobre la frecuencia del reloj se acepta un mínimo de 80 MHz en la CPU.

Se pretende ser lo más compatible posible con el lenguaje *Python* para permitirle transferir código con facilidad desde el escritorio de una

computadora personal hacia un microcontrolador o un sistema embebido.

MicroPython se desarrolla abiertamente en *GitHub* y el código fuente está disponible para que cualquier persona pueda contribuir con el proyecto[43].

El *firmware* para los diversos microcontroladores se puede encontrar en: <https://micropython.org/download/>, a continuación se muestra en la tabla 2.5 de los dispositivos más populares para trabajar con *MicroPython*:

Dispositivos más populares para el uso de MicroPython	Principales características
<i>Pyboard v 1.0</i>	MCU: STM32F411RE CPU: Cortex M4 de 96 MHz con unidad de punto flotante Memoria: ROM de 512 kb y RAM de 128 kb
<i>Pyboard v 1.1</i>	MCU: STM32F405RG CPU: Cortex M4 de 168 MHz con unidad de punto flotante Memoria: ROM de 1.024 kb y RAM de 192 kb Cuenta con un sensor de movimiento
<i>Micro:bit v1 BBC</i>	CPU: Arm Cortex-M0 32 de 16 MHz Memoria: ROM de 256 kb y RAM de 16 kb Cuenta con Bluetooth y un sensor de movimiento
<i>ESP32</i>	CPU: ESP32-D0WDQ6 32 bits de 240 Mhz Memoria: ROM de 520 Kb y RAM de 4 Mb Cuenta con Bluetooth y Wi-Fi
<i>ESP32-C3</i>	CPU: ESP32-D0W6 32 bits de 160 Mhz Memoria: ROM de 384 kb y RAM de 400 Kb Cuenta con Bluetooth y Wi-Fi
<i>ESP32-S2</i>	CPU: 32 bits de 160 Mhz de 240 Mhz Memoria: ROM de 2 Mb y RAM de 4 Mb Cuenta con Bluetooth y Wi-Fi
<i>ESP8266 with 512kiB flash</i>	CPU: Tensilica L106 de 52 Mhz Memoria: ROM de 512 Kb y RAM de 64 Kb Cuenta con Wi-Fi
<i>Arduino Nano 33 BLE Sense</i>	CPU: Arm Cortex-M4 de 64 Mhz Memoria: ROM de 1 Mb y RAM de 256 Kb Cuenta con micrófono, bluetooth, sensor de movimiento, temperatura, humedad y presión
<i>Arduino Nano RP2040 Connect</i>	CPU: Arm Cortex-M0 de 133 Mhz Memoria: ROM de 16 Mb y RAM de 264 Kb Cuenta con micrófono, bluetooth Wi-Fi y sensor de movimiento.
<i>Teensy 4.0</i>	CPU: Arm Cortex-M7 de 600 Mhz con unidad de punto flotante Memoria: ROM de 2048 kb y RAM de 1024 Kb
<i>Pico de Raspberry Pi</i>	CPU: Arm Cortex-M0 de 133 Mhz con unidad de punto flotante Memoria: ROM de 2 Mb y RAM de 264 Kb
<i>Nucleo F091RC St Microelectronics</i>	CPU: Arm Cortex-M0 de 48 Memoria: ROM de 512 Kb y RAM de 64 Kb

Tabla 2.5: Dispositivos más populares usados para el uso de *MicroPython*.

Existen muchas opciones de interpretes del lenguaje *Python* que permiten ejecutar nuestro código para obtener los resultados deseados, un ejemplo de esto es el entorno de desarrollo denominado Thonny el cual engloba editor, interprete y depurador, lo cual facilita la tarea de programación.

2.4.1. Thonny IDE

Un entorno de desarrollo integrado (IDE) es una aplicación de *software* que ayuda a los programadores a desarrollar código de *software* de manera eficiente. Aumenta la productividad de los desarrolladores al combinar capacidades como editar, crear, probar y empaquetar *software* en una aplicación fácil de usar[44].

Para hacer uso de *MicroPython* se utiliza el IDE Thonny el cuál cuenta con herramientas de programación para hacer uso de módulos de la empresa *Espressif* cómo lo son de la familia ESP32 y ESP8266, las cuales son las más comunes debido su bajo costo y altas prestaciones. Thonny es compatible con los sistemas operativos más populares del mercado: *Windows*, *Mac OS X* y *Linux*. El *software* Thonny tiene diversas características básicas tanto de los IDE cómo de Python, se incluye un depurador para corregir errores de sintaxis, por otro lado, se incluye acceso a la consola propia de *Python*, se muestra en la figura 2.13 una ventana del entorno gráfico de dicho IDE.

Es importante mencionar que este *software* ayudara en el proceso para la carga del firmware correspondiente [45].

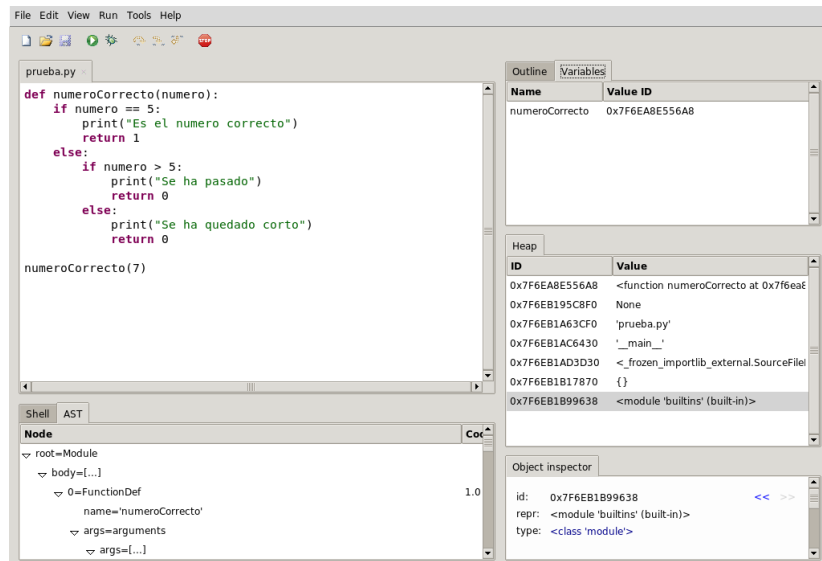


Figura 2.13: Ambiente gráfico del *software* Thonny.

2.5. Comunicación inalámbrica y acceso a los datos

El desarrollo de sistemas de monitoreo en tiempo real para las concentraciones de contaminantes en el ambiente interior no se limita a las conexiones de sensores y puertas de enlace. Es crucial identificar una tecnología de comunicación confiable para acceder a los datos registrados de forma remota.

La literatura revela que *Wi-Fi* es una de las opciones preferidas para la comunicación en los sistemas de monitoreo para la calidad del aire [46]. Sin embargo, otros investigadores también prefirieron usar la tecnología *ZigBee* y *Bluetooth* [13]. La principal limitación de *Wi-Fi* para mediciones en tiempo real es su mayor requisito de consumo de energía. Por otro lado, *Bluetooth* y *ZigBee* tienen una calificación alta por sus bajos requisitos de energía [47]. La comunicación *Wi-Fi* se establece generalmente mediante Protocolo IEEE802.11 b/n/g. Sin embargo, se recomienda IEEE 802.15.4 para la comunicación *ZigBee* [48]. Varios investigadores también trabajaron en el protocolo MQTT por su capacidad para admitir una implementación sencilla y un bajo consumo de energía [49]. El objetivo principal de estas tecnologías de comunicación es transferir datos registrados al servidor dedicado desde el sitio de destino. Se requieren tecnologías de comunicación para garantizar el almacenamiento y el acceso remoto de los datos.

Sin embargo, se tiene que considerar el almacenamiento de los datos generados por nuestro sistema. En general, la mejor recomendación para el monitoreo basado en *IoT* es la nube servidores que permiten un fácil acceso a los datos almacenados desde cualquier lugar, en cualquier momento. Además, también es posible agregar protección con contraseña para evitar el acceso no autorizado a datos ambientales para aplicaciones específicas de monitoreo comercial o residencial. *ThingSpeak* es un plataforma de *IoT* la cual se muestra en la figura 2.14 la visualización de datos que permite recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones *IoT*, es la solución de almacenamiento preferida para los sistemas de monitoreo de la calidad del aire en ambientes cerrados. *ThingSpeak* permite el almacenamiento de datos de cuatro canales diferentes donde cada canal puede contener 8 parámetros únicos. Esta es una solución eficiente para grandes proyectos de monitoreo [50], en un apartado posterior se hablara un poco más sobre esta importante plataforma de *IoT* la cual será empleada en la visualización y obtención de datos en este trabajo de tesis.

Existen otras opciones de plataformas conectadas a la nube cómo lo son *Ubidots*, *Grafana* y *Adafruits*, las cuales cuentan con tablas de visualización de datos, denominadas *IAQ Dashboards*.

Una alternativa del almacenamiento de los datos es el uso de servidores locales, almacenamiento interno móvil y tarjetas SD. Los datos registrados deben ser accesibles para los usuarios finales y, para ello, también debemos trabajar en una selección de los métodos de consulta de datos más fiables [13].

El número de usuarios de teléfonos inteligentes aumenta con los años, es más fácil proporcionar acceso en tiempo real a datos ambientales a través de aplicaciones móviles [51]. Sin embargo, el uso de servidores web y portales es otra recomendación popular para mostrar las características de la calidad del aire de una manera más rápida y eficaz. También existe la opción de usar una pantalla LCD o de tecnología *Oled* con el *hardware* del sistema de monitoreo para mostrar las concentraciones de contaminantes en el sitio [13].



Figura 2.14: Dashboard presentado en el *software ThingSpeak*.

Sin embargo, la mayoría de los sistemas de la era moderna se hacen compatibles con las aplicaciones móviles para brindar información clara de una manera más rápida y a distancia con el uso de dispositivos que se conecten a internet y puedan hacer consultas de las mediciones generadas. Es importante considerar la seguridad de los datos generados, es por ello que la gran mayoría de los sistemas basados en la web se pueden proteger con credenciales de inicio de sesión para obtener actualizaciones de parámetros [13].

2.5.1. Protocolo de comunicación MQTT

El protocolo MQTT se ha convertido en uno de los principales pilares del *IoT* por su sencillez y ligereza. Ambos son condicionantes importantes dado que los dispositivos *IoT*, a menudo, tienen limitaciones de potencia, consumo y ancho de banda.

MQTT *Message Queuing Telemetry Transport* es un protocolo de mensajería simple diseñado para dispositivos con ancho de banda limitado y alta latencia. El modelo de este protocolo se visualiza en la figura 2.15, el cuál se basa en el paradigma publicador-suscriptor en el que un broker actúa como intermediario encargándose de direccionar los mensajes con el uso de un topic común [52]. El protocolo trabaja sobre TCP/IP u otros protocolos que permitan conexiones bidireccionales, ordenadas y sin pérdidas. Su principal objetivo es minimizar los requerimientos de ancho de banda de red y los recursos de los dispositivos que lo usan,

manteniendo cierto grado de habilidad, lo que hace que su uso en *IoT* o conexiones machine-to-machine (M₂M) sea muy conveniente.

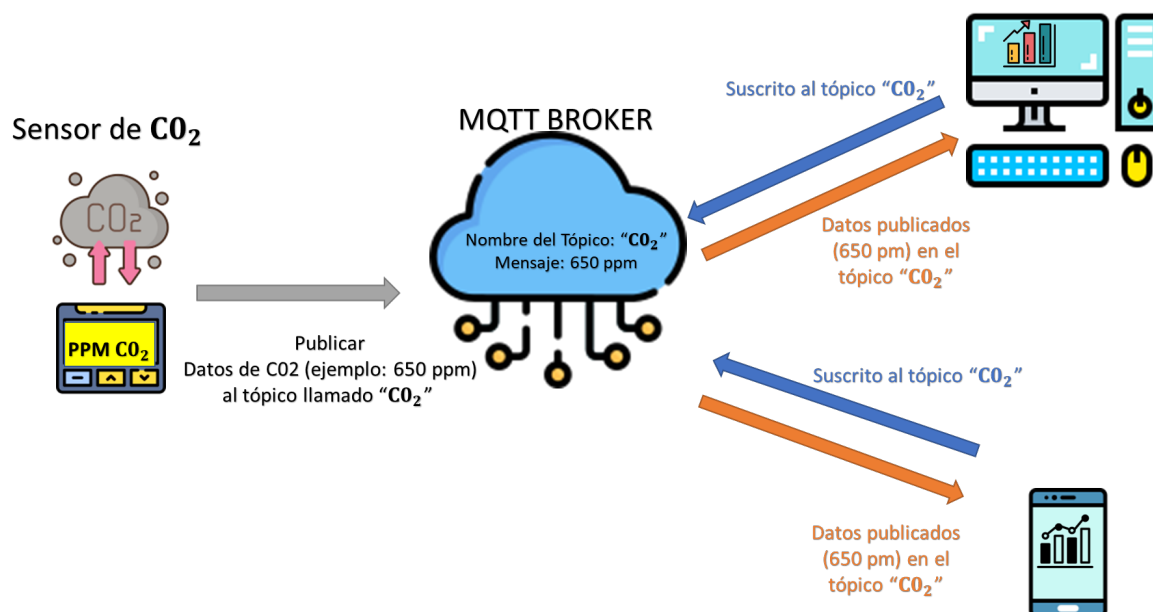


Figura 2.15: Modelo de comunicación MQTT.

2.5.2. Características principales de MQTT

La característica principal en el modelo MQTT es que sigue la arquitectura de comunicación cliente/servidor normalmente con una topología en estrella, con un nodo central que funciona de servidor y con una capacidad de hasta 10.000 clientes [53]. A continuación se describen los componentes de *software* principales que aparecen en un escenario MQTT [52]:

- **Broker:** este componente actúa como servidor encargándose de la transmisión de mensajes con los clientes y la gestión de la red. También mantiene activo el canal con los clientes respondiendo a los mensajes periódicos que estos envían.
- **Cliente:** el cliente puede tener funciones de publicador y de suscriptor al mismo tiempo.
 - **Publicador:** actúa como cliente y se encarga de transmitir información al broker sobre un determinado topic.
 - **Suscriptor:** actúa como cliente y se encarga de recibir infor-

mación del broker sobre un determinado topic.

- **Mensaje:** unidad de datos o información que se transmite o recibe sobre un topic.
- **Topic:** el tópicos al que los clientes pueden suscribirse para recibir información sobre ese asunto o publicarla. Es similar a una cola de mensajes, pero los mensajes pueden almacenarse hasta que sean consumidos y se pueden distribuir a varios clientes.

2.6. Plataforma *IoT ThingSpeak*

ThingSpeak es un servicio de plataforma de análisis de *IoT* en las cuales se tiene permitido agregar, visualizar y analizar flujos de datos en la nube. Es posible enviar datos a *ThingSpeak* desde dispositivos como lo son computadoras, teléfonos inteligentes o incluso sistemas embebidos con la capacidad de conexión a internet. Se puede crear una visualización instantánea de datos, enviar alertas o crear gráficos a partir de los datos almacenados, un ejemplo de la visualización de dichos datos se puede observar en la figura 2.14.

2.6.1. Características principales de *ThingSpeak*

- **Envío de datos del sensor de forma privada a la nube:**
ThingSpeak permite que sensores, instrumentos y sitios web envíen datos a la nube, donde se almacenan en un canal privado o público. *ThingSpeak* almacena datos en canales privados de forma predeterminada, los canales públicos se pueden usar para compartir datos con otros usuarios. Una vez que los datos están en un canal de *ThingSpeak*, se puede analizar, visualizar y calcular nuevos datos o interactuar con redes sociales, servicios web y/o otros dispositivos con conexión a la nube [54].
- **Análisis y visualización de datos con Matlab:**
 - Es posible convertir, combinar y calcular nuevos datos.
 - Programar los cálculos para que se ejecuten en determinados momentos.

- A partir de los datos generados se puede comprender visualmente la relación de información haciendo uso de funciones de trazado integradas en Matlab.
 - Combinar datos de varios canales de *ThingSpeak*, para crear un análisis más sofisticado.
- **Activar una reacción:**
- Es posible activar una acción a los datos ingresados al canal a través de notificaciones visuales, envío de mensajes a través alguna red social como lo es *Twitter* o la activación de dispositivos de *hardware*.

3. Desarrollo

En este capítulo se hace una descripción detallada de los elementos usados y la integración para el funcionamiento del sistema embebido, el desarrollo del proyecto se dividirá en los bloques fundamentales mostrados en la figura 3.1, posteriormente se describe cada bloque, los elementos que lo conforman y las pruebas realizadas.

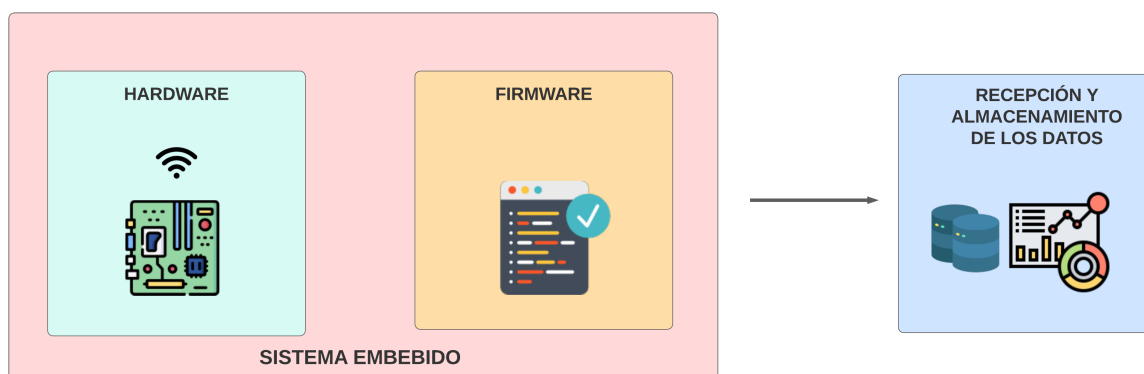


Figura 3.1: Diagrama de bloques del desarrollo del sistema.

3.1. Desarrollo del *hardware*

Para las primeras pruebas que se realizaron del sistema se procede a realizar un armado en placas de experimentación para el prototipado electrónico (protoboard), es importante mencionar que el sensor de detección de partículas de CO_2 , con el cual se realiza esta primer prueba es el sensor de gas MQ135, un sensor que emplea tecnología electroquímica, esto debido a que inicialmente no se contaba con el sensor definitivo, más adelante se reemplaza dicho sensor por el de tecnología NDIR en específico el SenseAir S8.

Antes del armado de manera física del circuito, se realiza un diagrama esquemático del sistema, el cual se muestra en la sección de apéndices en la figura 6.1, se muestran los diferentes componentes del circuito y las conexiones de alimentación y de señal eléctrica entre los dispositivos haciendo uso de la hoja de datos de cada componente y/o módulo usado.

3.1.1. Pruebas de funcionamiento en placa de prototipo electrónico

Se desarrolla el sistema en dos placas de prototipado electrónico de 830 puntos y se hace uso de componentes de tecnología de agujeros pasantes (THT del inglés *Through-Hole Technology*) y tres módulos que emplean de la misma manera la tecnología antes mencionada.

El prototipo esta conformado por 5 bloques fundamentales:

- **Sensor de CO₂**: Módulo que incluye un sensor de gas el cual detectará las partes por millón del CO₂ en el ambiente, en este caso se hará uso del MQ135 debido a su bajo costo y sus altas prestaciones para la detección de partículas nocivas para el ser humano, este sensor cuenta con la capacidad de detección de 10-1000 ppm de CO₂, para fines de prueba en la integración del primer prototipo es una buena elección.
- **Tarjeta de desarrollo ESP 32**: La cual incluye un módulo ESP-32, el cual cuenta con un SoC que contiene el firmware correspondiente y la etapa electrónica necesaria para enviar información de forma inalámbrica a través de Wi-Fi. Para el sistema electrónico de control, se ha seleccionado el módulo ESP-WROOM-32.
- **Pantalla electrónica** : Dispositivo el cual notifica de forma visual los datos obtenidos a través del dispositivo que exhibe el resultado en una pantalla de tecnología OLED SSD130 ya que tiene ventaja de tener un consumo bajo debido a que solo se enciende el píxel necesario y no requieren de luz de fondo y tienen una mejor visibilidad en ambientes luminosos, como bajo el sol en comparaciones de otros exhibidores.
- **Reproductor de audio**: Adicionalmente el sistema embebido cuenta con notificaciones mediante voz, para advertir las condiciones en el cual se encuentra la calidad del aire y el riesgo que implica, se hace uso de un *DFplayer mini* reproductor mp3 con salida de audio a unas bocinas con plug 3.5 mm, este dispositivo fue seleccionado ya que es un módulo electrónico de fácil implementación y con la capacidad de reproducción de pistas de audio en los formatos más comunes, cómo lo son MP3, WMA y WAV.
- **Etapa de potencia**:Módulo que contiene un relevador con un vol-

taje de alimentación de 5 VDC, su activación se realiza a partir de una etapa de acoplamiento conformada por un transistor pnp el cual desempeña la función de interruptor, la carga del relevador es conectada al pin NC (Normalmente cerrado), en este caso se hará la conexión de un ventilador con una alimentación de 127 VAC. La activación de la carga será un auxiliar para la circulación del aire debido a una alto nivel de ppm de CO₂ en el ambiente.

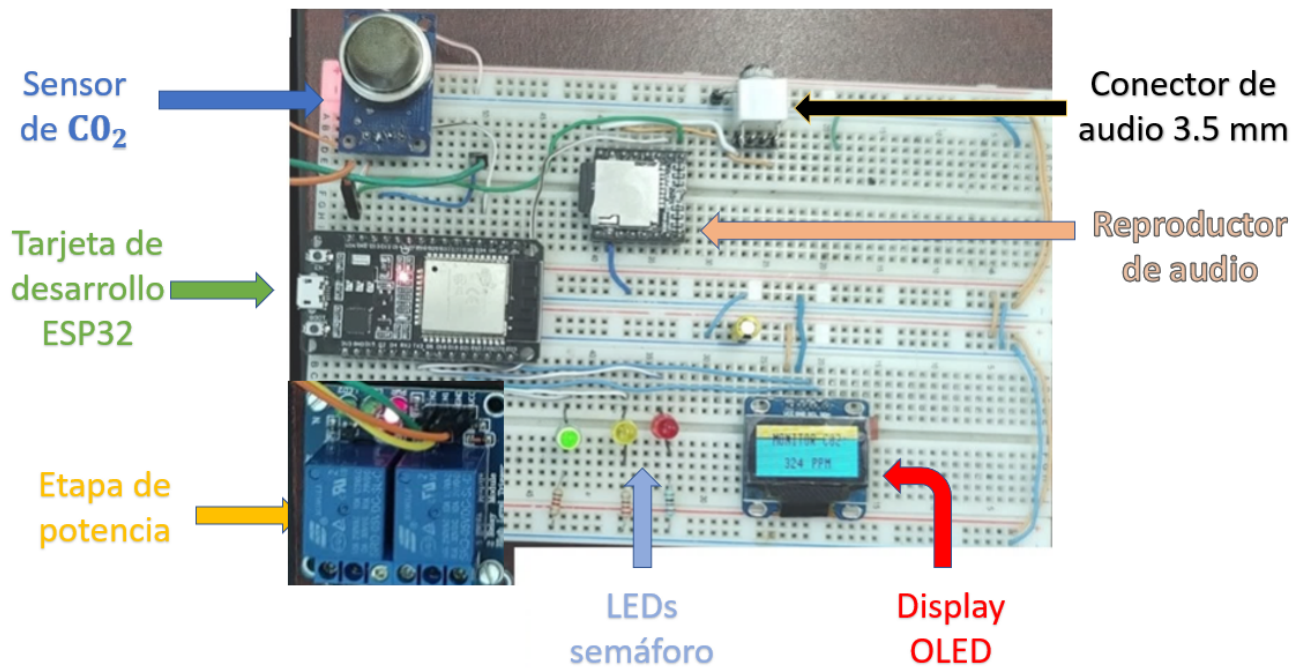


Figura 3.2: Prototipo sistema embebido.

El sistema armado es presentado en la figura 3.2, se tiene una excelente respuesta de integración entre todos sus componentes, para efectos de pruebas y depuración de *hardware* es muy útil, incluso las pruebas de conexión a internet fueron posibles de manera satisfactoria, lo cual será mostrado en las secciones posteriores.

Sin embargo para la realización de la PCB es importante hacer una prueba sin los componentes de programación contenidos en la tarjeta de desarrollo ESP-32, esto debido a que dicha etapa no es necesaria es nuestro sistema embebido, y el hecho de descartarla es una ventaja en el ahorro de energía del sistema, por consiguiente el hecho de integrar el módulo ESP-32 a una placa base y verificar un primer diseño de un prototipo, es importante ya que esta etapa será la cual envíe información a través de internet.

Es importante mencionar que lo mostrado a continuación es una PCB de prueba que será utilizada para validar la correcta programación del ESP-32, posteriormente se hará una placa electrónica que contenga todos los dispositivos que formaran al sistema embebido con todas las funciones y características mencionadas en los objetivos específicos.

3.1.2. Diseño de PCB ESP-32 WROOM en una placa base

Este capítulo presenta los puntos clave del diseño con el módulo ESP-32 WROOM se diseña un prototipo donde se implementa el módulo en una placa base, considerando la guía de diseño de *hardware* proporcionada por la empresa *Espressif Systems* [55]. Si bien el alto nivel de integración del módulo a usar hace que el proceso de diseño de PCB sea simple, el rendimiento del sistema depende en gran medida de los aspectos de diseño del sistema, es importante mencionar que el diseño del esquemático y la placa electrónica se efectúa en el *software* de desarrollo electrónico *Eagle* de la empresa *Autodesk* con la versión 9.6.2.

Diseño del esquemático

Se realiza un esquemático del sistema en el que contiene ocho bloques, podemos observar dicho diagrama en la sección de apéndices en la figura 6.2, los bloques se describen a continuación:

- **LDO (Regulador de voltaje):** Etapa de regulación haciendo uso del regulador de voltaje AMS1117 con encapsulado SOT-223, diseñado para proporcionar una corriente de salida de 1 A. y una entrada máxima de 15 V. con una salida de 3.3 V. se diseña el circuito con los capacitores de entrada y salida necesarios para la estabilidad, y mejorar la respuesta transitoria.
- **Pulsadores:** Se cuenta con dos botones los cuales están conectados a los siguientes pines del módulo ESP-32:
 - **Boot:** Botón de montaje superficial el cual está conectado al pin número 25 del módulo ESP-32, el cual al presionarlo durante tres segundos hará la carga del firmware.
 - **Reset:** Botón de montaje superficial el cual está conectado directo al pin número 3 del módulo ESP-32, el cual al presionarlo

deshabilita al chip.

■ **Leds:**

- **Led Power:** Indicador de montaje superficial, conectado a la salida del regulador de voltaje con una resistencia de 1 K Ω , para indicar la energización del sistema en general.
- **Led GPIO2:** Indicador de montaje superficial, conectado a la salida digital número 2 del módulo ESP-32 con una resistencia de 1 K Ω , para un uso de propósito general.

■ **Módulo ESP-32 WROOM:** Módulo electrónico el cual integra *Wi-Fi* y *Bluetooth*, ideal para desarrollar productos de IoT, implementado en un dispositivo de montaje superficial, el núcleo de este módulo está compuesto por el SoC ESP-32-D0WDQ6 en el cual se hará la carga del *firmware* utilizado en este proyecto.

■ **Conectores GPIO:**

- **Sensor MQ135:** Conector tipo hembra con una separación de 2.54 mm. en el cual se conectara el sensor MQ135 a la entrada analógica en el pin 34 del módulo ESP-32, se cuenta la conexión necesaria de voltaje y tierra correspondientes para la correcta energización del sensor.

■ **Transistores de *autoreset*:** Haciendo uso de transistores bipolares NPN con encapsulado SOT-23, se encuentran conectados las bases a resistores de 10 K Ω y a su vez a los pines DTR Y RTS de nuestro dispositivo convertidor TTL-USB (del cual se hablara mas adelante), haciendo las siguientes combinaciones en función a los modos en los cuales queremos que se programe el módulo ESP-32:

- Si DTR esta en un nivel lógico bajo, RTS alterna su nivel lógico de alto a bajo y se reinicia el módulo ESP-32 al modo de ejecución.
- Si RTS esta en un nivel lógico alto, DTR alterna su nivel lógico de bajo a alto y se reinicia el módulo ESP-32 al modo *Bootloader*.

■ **Conector para comunicación I²C:** Conector tipo hembra con una separación de 2.54 mm. en el cual se conectara una pantalla de tecnología OLED SSD130 a los correspondientes pines de salida

del módulo ESP-32 con comunicación I²C, SDA al pin numero 34 y SCL al pin 36, se cuenta con la conexión necesaria de voltaje y tierra correspondientes para la correcta energización de la pantalla.

- **Conector TTL-USB:** Conector tipo hembra con una separación de 2.54 mm. en el cual se hace la conexión de un módulo electrónico CP2102 el cuál es un convertidor USB-TTL, para realizar la carga del *firmware* y los programas hacia el módulo ESP-32.

Diseño de placa electrónica

Posteriormente a la realización del esquemático del sistema se efectúa el diseño de la placa electrónica (PCB), se consideran las siguientes especificaciones en el diseño realizado:

- La capa superior es utilizada para pistas y componentes
- En la capa inferior no se debe colocar ningún componente electrónico y se debe colocar pistas mínimas idealmente debería ser un plano de GND.
- El módulo ESP-32 se coloca lo más cerca posible del borde de la placa base para aumentar el rendimiento de la antena.
- El área aledaña a la antena debe mantenerse libre de cobre y componentes electrónicos al menos 15 mm.
- Deben existir al menos dos vías de alimentación para cruzar de la capa inferior a la superior.
- El ancho de la vía de alimentación debe ser igual o mayor al ancho de las pistas de alimentación.
- Se requiere de una almohadilla térmica en el centro del módulo.
- Las pistas de alimentación de 3,3 V, resaltadas en deberán tener un ancho de la principal superior a 20 mil (milésimas de pulgada).
- El ancho de las pistas de potencia para el pin 3 y pin 4 debe ser al menos 20 mil.
- El ancho de las pistas de energía debe estar dentro del rango de 12 a 15 mil.

Se obtiene una PCB de dos capas de cobre en ambos lados del sustrato, se observa en la figura 3.3 el resultado de dicha placa.

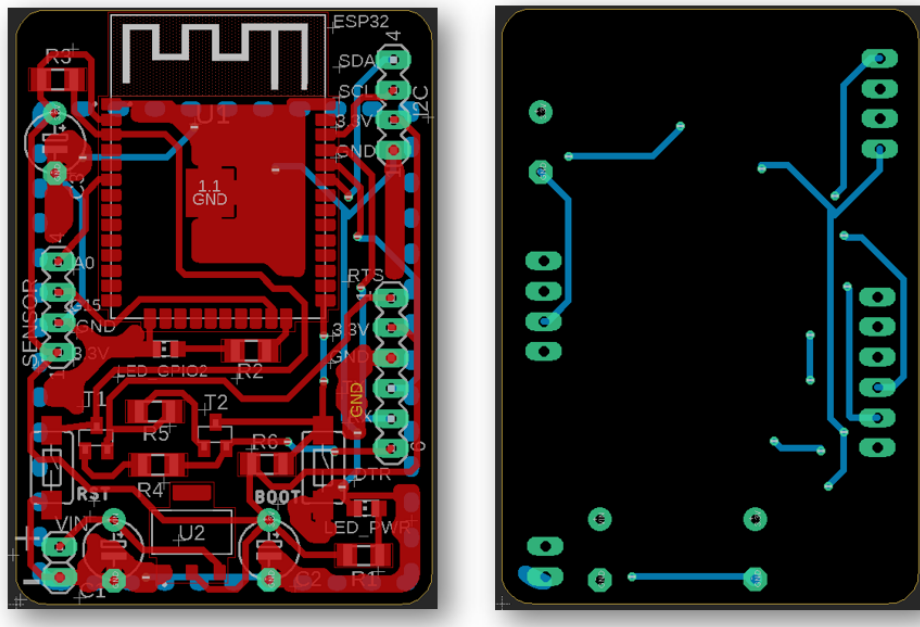


Figura 3.3: Diseño de la PCB ESP-32 en una placa base, izquierda: vista superior, derecha: vista inferior.

3.1.3. ECAD de la ESP32-WROOM en una placa base

Casi todos los diseños electrónicos se encuentran dentro de un producto mecánico. Con la integración de ECAD entre *Eagle* y *Fusion 360*, es más fácil producir una carcasa para el sistema electrónico y verificar los ajustes necesarios para validar los resultados en un producto final.



Figura 3.4: Colocación de componentes electrónicos modelados en 3D.

Es por ello que se realiza la correspondiente simulación de la PCB en 3D considerando los siguientes puntos:

- Verificar una correcta colocación de planos de tierra, vías, almohadillas térmicas, pistas para las conexiones eléctricas, antena y etiquetas de componentes.
- Verificar las medidas de los componentes y su respectivo *footprint*.
- Analizar y validar las posiciones finales de los componentes electrónicos sin afectar el espacio entre ellos y los bordes de la tarjeta. antes de fabricar la PCB física.

Se procede a obtener todos los archivos de los componentes electrónicos colocados en la PCB como se muestra en la figura 3.4, en un formato *.step* ya que se utiliza con frecuencia para la impresión y el modelado en 3D, algunos son creados por nuestra cuenta desde el *software Eagle* y otros más son descargados de paginas web de modelado 3D como lo son *Grabcad* en su portal oficial de internet: <https://grabcad.com/dashboard>.

Al finalizar con el montaje de los componentes electrónicos es posible exportar el documento de diseño de la placa electrónica creada en el *software Eagle* hacia el *software Fusion 360* como se muestra en la figura 3.5.

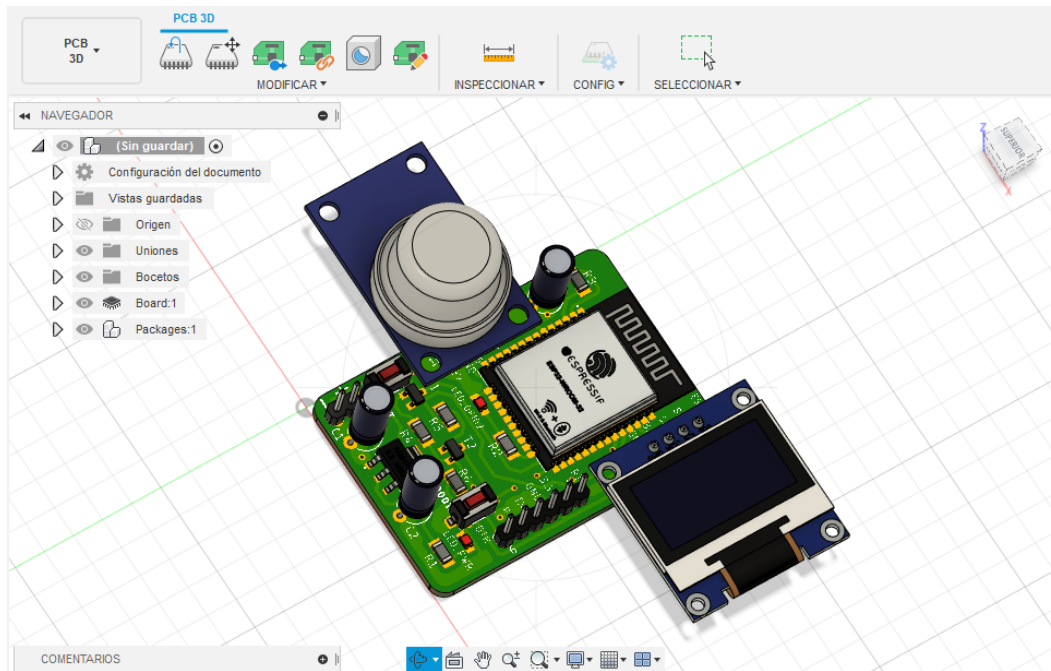


Figura 3.5: ECAD desarrollado en el *software Fusion 360* del ESP32-WROOM en una placa base.

Se generan algunas imágenes de la placa electrónica en diferentes perspectivas para validar la posición de los componentes electrónicos y la colocación de los mismos, como se puede observar en la figura 3.6.

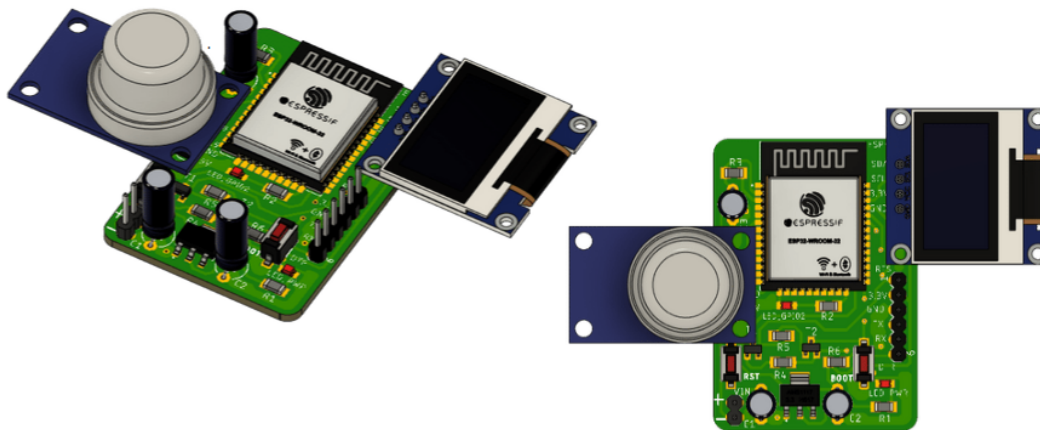


Figura 3.6: Imágenes generadas de la placa electrónica en 3D.

3.1.4. Fabricación y ensamblaje de PCB ESP32 WROOM

La fabricación de esta placa prototipo se realiza con un material denominado FR-4 estándar con un espesor de de 2 mm. doble capa de

cobre recubierta con una pintura de serigrafía color amarilla en la capa de *soldermask*, la cual contendrá los nombres y las etiquetas de los componentes. El resultado final se presenta en la figura 3.7.

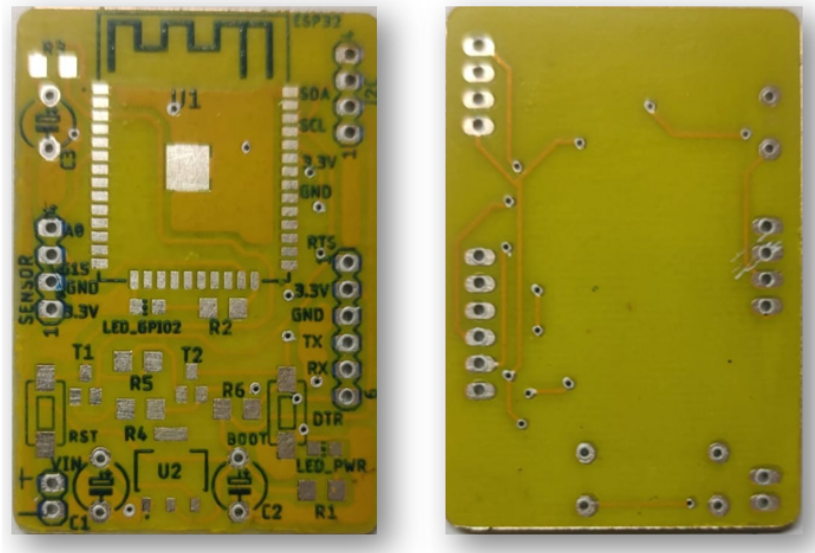


Figura 3.7: PCB sin ensamblar.

El montaje de los componentes se realiza de forma manual y con una temperatura de 300° centígrados, los componentes montados en su mayoría son de tecnología smd (montaje superficial) con empaquetados 1206, 0805 y SOT-223, algunos dispositivos son de tecnología de agujeros pasantes (THT). El resultado del ensamble se visualiza en la figura 3.8.

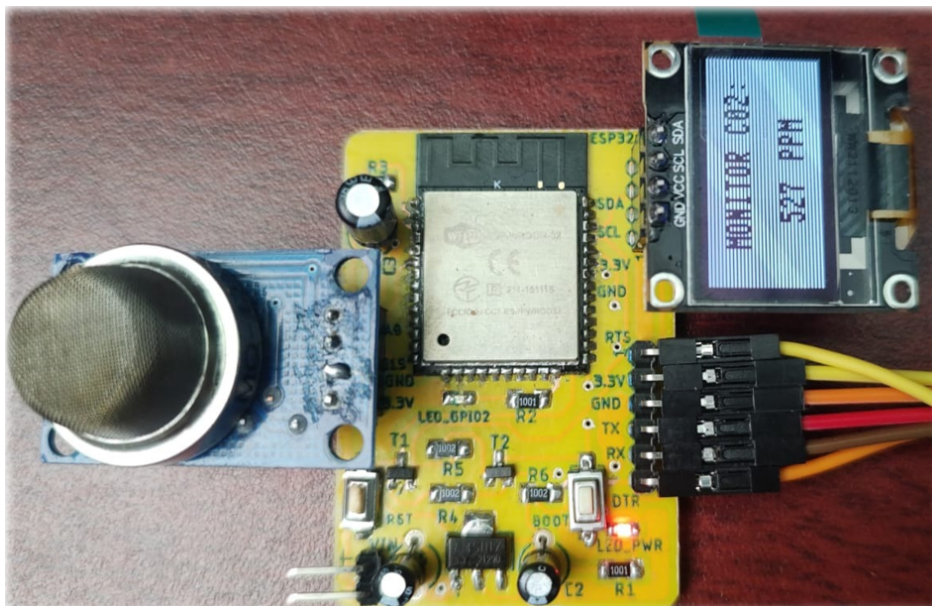


Figura 3.8: PCB ensamblada.

3.2. Programación del *firmware* de la placa base ESP32-WROOM

El código *firmware* basado en *Python* fue desarrollado teniendo en cuenta todos los componentes del sistema embebido y sus características necesarias para una correcta operación.

Se hizo uso del entorno de desarrollo denominado *Thonny*, para la carga del *firmware* de *MicroPython* a este proceso se le conoce como *flasheado* del SoC en este caso del ESP32-WROOM.

El *firmware* usado fue la versión 1.14 publicada el 2 de febrero del 2022 en el repositorio oficial de *MicroPython*: <http://www.micropython.org/download/esp32> se elige esta versión ya que es un *firmware* depurado y probado ampliamente por la comunidad de desarrolladores, además cuenta con soporte del *Bluetooth* y las características necesarias para establecer conexión inalámbrica vía *Wi-Fi*.

El procedimiento de la carga del *firmware* en el *IDE* de *Thonny* se muestra en la figura 3.9.

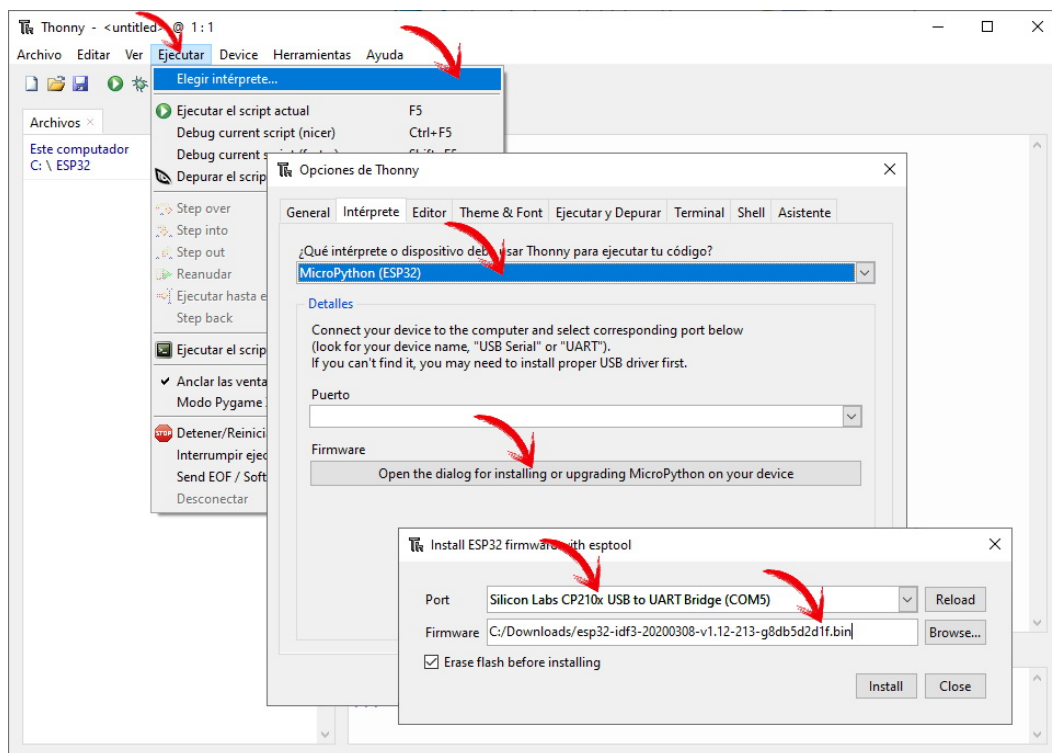


Figura 3.9: Ventanas generadas para la instalación del *firmware* de *MicroPhyton* en el *software* *Thonny*.

La instalación inicia al presionar brevemente el botón de *RESET* en la placa de desarrollo del módulo ESP-32.

3.3. Desarrollo del *firmware* de la placa base ESP32-WROOM

La realización del código implementado en *MicroPhyton* se realiza con la carga de tres archivos en *Python* con extensión *.py*, los archivos cargados se describen a continuación:

- ***main.py***: Archivo principal donde ejecuta el punto de entrada del programa, se invocan los archivos vistos anteriormente y se lleva a cabo el algoritmo mostrado en la figura 3.10.
- ***boot.py***: Este archivo se ejecuta en cada arranque del SoC ESP-32(incluido el arranque en modo de energía bajo). Es proporcionado de forma automática al ser cargado el *firmware* de *MicroPhyton*.
- ***dfplayermini.py***: Archivo que contiene la biblioteca necesaria para el funcionamiento del módulo reproductor *Df player mini*, se defi-

ne la comunicación serial, volumen de la reproducción de audio y los comandos hexadecimales necesarios para la inicialización del módulo, se visualiza la biblioteca usada de *MicroPhyton* en el repositorio de github <https://n9.cl/iwahg>.

- **ssd1306.py:** Archivo que contiene la biblioteca necesaria para el funcionamiento de la pantalla OLED SSD1306 con comunicación I²C, se define el contraste y la resolución de la pantalla así como el protocolo para la comunicación, la biblioteca usada para código en *MicroPhyton* se encuentra en el apartado de apéndices en el repositorio de github <https://n9.cl/yairromero>.

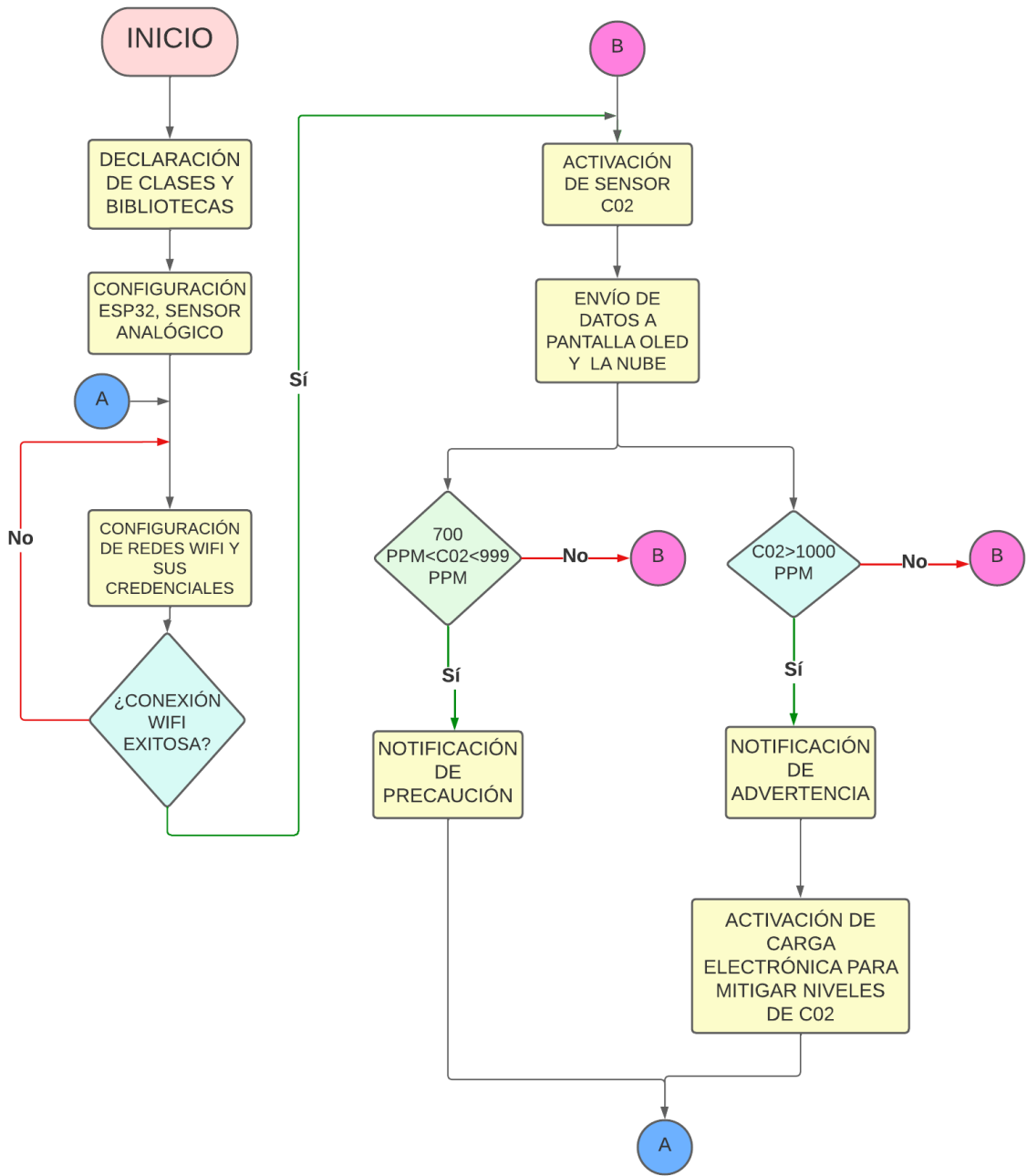


Figura 3.10: Algoritmo implementado en el ESP-32.

3.4. Desarrollo de una *dashboard* en *software IoT: ThingSpeak*

3.5. Pruebas de conexión hacia la nube

Se procede a realizar pruebas de conexión a internet haciendo uso de las herramientas mostradas en la figura 3.11.

En las herramientas de *software* se utiliza *ThingSpeak* para crear una *dashboard* y de esta forma visualizar los datos obtenidos. Para la carga y programación del *firmware* de nuestra tarjeta de desarrollo ESP-32 se hace uso de *MicroPython* el cual hará la conexión con *ThingSpeak* mediante el protocolo de comunicación MQTT.



Figura 3.11: Recursos usados para pruebas de conexión a Internet.

3.5.1. Creación de un tablero en el *software ThingSpeak*

Para realizar las primeras pruebas de envío del prototipo es necesario crear una cuenta en la página oficial del *software*; <https://thingspeak.com>, esto es necesario para generar un tablero donde se presente la información enviada a la nube.

Name ↕	Created ↕	Updated ↕
🔒 Medidor de prueba	2022-06-20	2022-08-25 21:12

Figura 3.12: Canal generado para las primeras pruebas de conexión hacia la nube.

Se generó para esta etapa un canal denominado "Medidor de prueba" mostrado en la figura 3.12.

Se realiza la creación de una *dashboard* la cuál se muestra en la figura 3.13 donde se presenta la siguiente información:

- Gráfico de los niveles de CO₂ obtenidos.
- Valor numérico en ppm de los niveles de CO₂ obtenidos
- Activación de una lampara para advertir una condición de riesgo de contagio al registrarse valores de CO₂ > 700 ppm.
- Localización del sensor de CO₂.

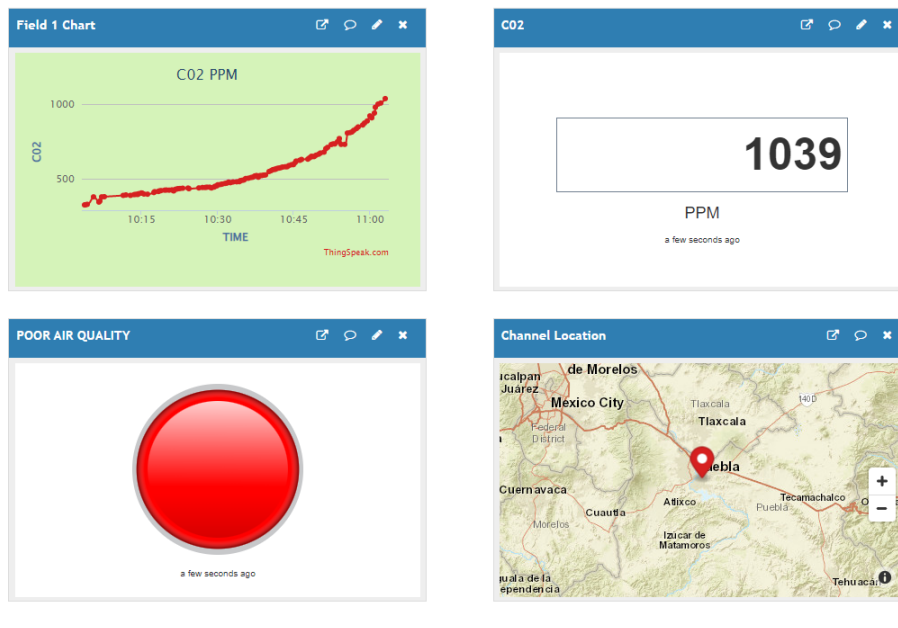


Figura 3.13: *Dashboard* generada en el *software* *ThingSpeak*

Del gráfico observado en la figura 3.13, existe la posibilidad de generar una tabla con los valores registrados, ya que esta información se puede exportar y presentarse en una tabla con la extensión .CSV. Los datos se almacenará en la cuenta vinculada al *software* *Thingspeak*, don-

de se tiene acceso al panel de control y a la tabla de datos generadas, se visualizan los datos en la tabla 3.1.

Created at	Entry_id	PPM CO ₂
2022-08-25T16:13:34-05:00	1	320
2022-08-25T16:13:55-05:00	2	321
2022-08-25T16:14:16-05:00	3	315
2022-08-25T16:14:50-05:00	4	315
2022-08-25T16:15:13-05:00	5	317
2022-08-25T16:15:44-05:00	6	325
2022-08-25T16:16:12-05:00	7	324
2022-08-25T16:16:34-05:00	8	326
2022-08-25T16:16:50-05:00	9	328
2022-08-25T16:17:08-05:00	10	328
2022-08-25T16:17:31-05:00	11	330
2022-08-25T16:17:51-05:00	12	329

Tabla 3.1: Formato de tabla generada por las mediciones obtenidas

A continuación se presenta el diseño de la tarjeta electrónica del sistema embebido propuesto, se integran las fases electrónicas de las secciones anteriormente vistas, añadiendo etapas nuevas para la optimización y el mejoramiento del mismo, se describen las etapas añadidas en las siguientes secciones.

3.6. Versión final del *hardware* del sistema integrado.

En este punto del trabajo de tesis planteado, se tiene pruebas del sistema electrónico propuesto en 1.1, se ha identificado un funcionamiento conforme a lo esperado. Se puede observar en las secciones anteriores que las pruebas realizadas tanto en placa de prototipo electrónico y en la primera versión de PCB diseñada con el esquema originalmente presentado, muestra un funcionamiento satisfactorio.

Se propone la adición de 4 bloques electrónicos con módulo de conversión de corriente alterna a corriente directa y un módulo de cargador de baterías de tecnología LiPo, (Polimeros de litio), la adición de estos módulos se observa en la figura 3.14, los cuáles podrán aumentar las funcionalidades planteadas originalmente en este tema de tesis. Se propone la adición de 4 bloques electrónicos con módulo de conversión de corriente alterna a corriente directa y un módulo de cargador de baterías

LiPo, la adición de estos módulos podrán aumentar las funcionalidades planteadas originalmente en este tema de tesis.

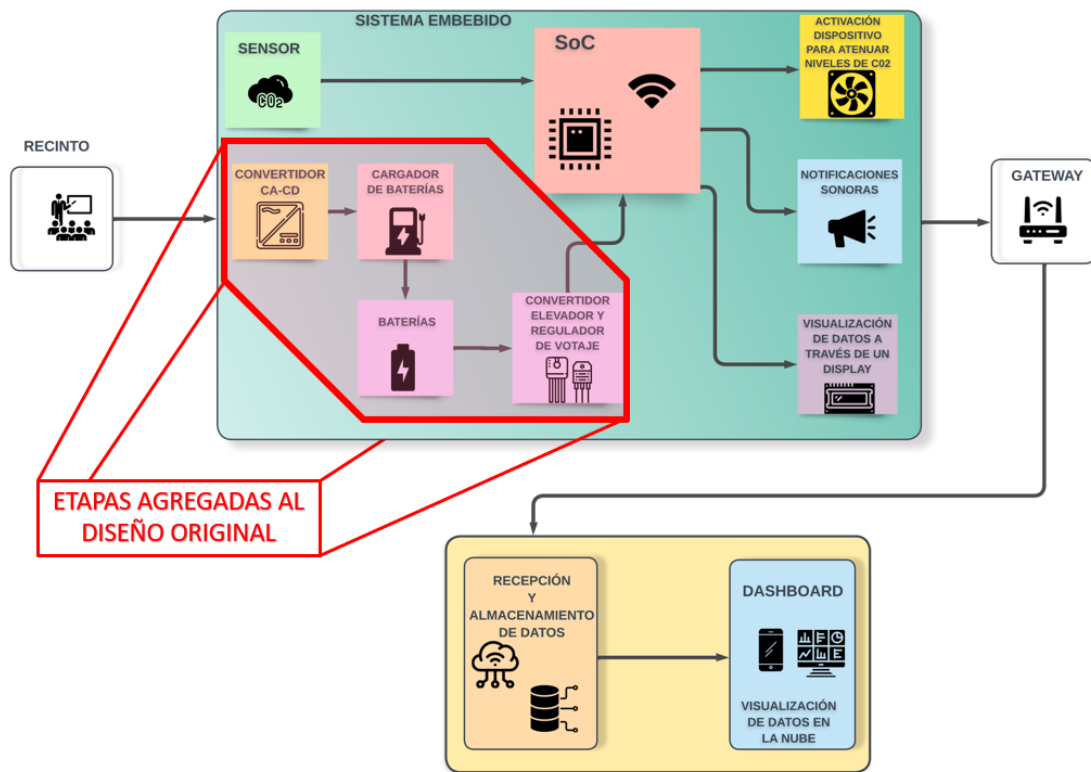


Figura 3.14: Diagrama de bloques de la versión final del sistema embebido

A continuación se describen con mayor detalle estos módulo:

3.6.1. Módulo convertidor AC-DC:

La fuente de alimentación del módulo de serie ultra-pequeña de 5 W con matrícula HLK-5M05, es una fuente de alimentación de conversión AC-DC de pequeño volumen y alta eficiencia diseñada por Shenzhen Hi-Link Electronics Co. Ltd. Tiene las ventajas del rango de voltaje de corriente alterna diseñado para la normativa Europea y Americana de 250 volts a 120 volts de corriente alterna respectivamente, aumento de baja temperatura, bajo consumo de energía, alta eficiencia, alta confiabilidad y aislamiento de alta seguridad. Ha sido ampliamente utilizado en hogares inteligentes, control de automatización, equipos de comunicaciones, instrumentación y otras industrias[56]. Dicho módulo tendrá una conexión con un conector bornera donde ingresaran el voltaje en

corriente alterna, posteriormente se hará la conexión hacia las entradas del módulo, para obtener una salida de 5 volts de corriente directa que en paralelo se conecta un capacitor de aluminio de 220 uf a 25 volts, esto es para reducir el voltaje de rizado de salida del modulo de 50 mV a 30 mV. La figura 3.15 muestra el esquemático del circuito realizado, teniendo en cuenta las especificaciones del fabricante y la configuración típica propuesta en la hoja de datos del módulo [56].

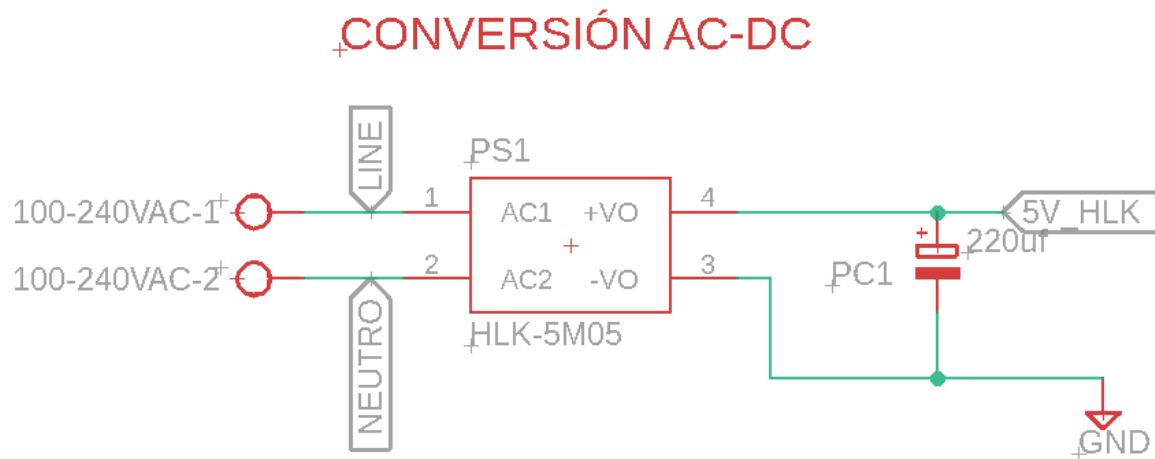


Figura 3.15: Esquema de implementación del módulo de conversión de AC-DC

3.6.2. Cargador de baterías de tecnología LiPo

Para la carga de baterías de tecnología LiPo se hará uso del circuito integrado TP4056 en empaquetado SMD SOP, es un cargador de batería de iones de litio para una batería de una celda, que protege la celda de carga excesiva o insuficiente. Tiene dos salidas de estado que indican carga en curso y carga completa. También tiene una corriente de carga programable de hasta 1 Ampere, el voltaje de carga se fija con una resistencia de 1.2 KΩ lo cual indica una carga de batería a 1000 miliamperes y un voltaje fijo de 4.2 volts. El cargador tiene la funcionalidad de finalizar automáticamente el ciclo de carga cuando la corriente de carga cae a $\frac{1}{10}$ del valor programado para después alcanzar el voltaje de flotación final (El voltaje de flotación es aquel que se mantiene la batería después de haber sido completamente cargada para mantener esa capacidad mediante la compensación de la auto descarga de la batería) [57].

El esquemático propuesto para la carga de la batería se muestra

en la figura 3.16, una configuración típica (sugerida por el fabricante del dispositivo [57]). El diagrama de bloques también muestra el pinout TP456 para el dispositivo SMD de 8 pines.

Adicionalmente se incorpora el circuito integrado DW01A el cual proporciona protección a la baterías de iones de litio de una sola celda y protege la celda de carga excesiva o insuficiente, conexión inversa y cortocircuito, se hace uso de la referencia del fabricante [58].

El circuito integrado DW01A requiere un transistor de tecnología mosfet dual externo para controlar la línea de tierra a la batería. Al usar el mosfet, la línea de tierra se desconecta al existir una condición de error (cortocircuito y/o sobre descarga). Esto aísla la batería de litio hasta que se retira la carga (en caso de cortocircuito), se hace uso del transistor F28205 donde se incluye dos transistores de canal tipo N y empaquetado SOT 23-6, al tener integrado todo en un circuito integrado smd, se hace uso de las referencias de diseño de los fabricantes [58][59].

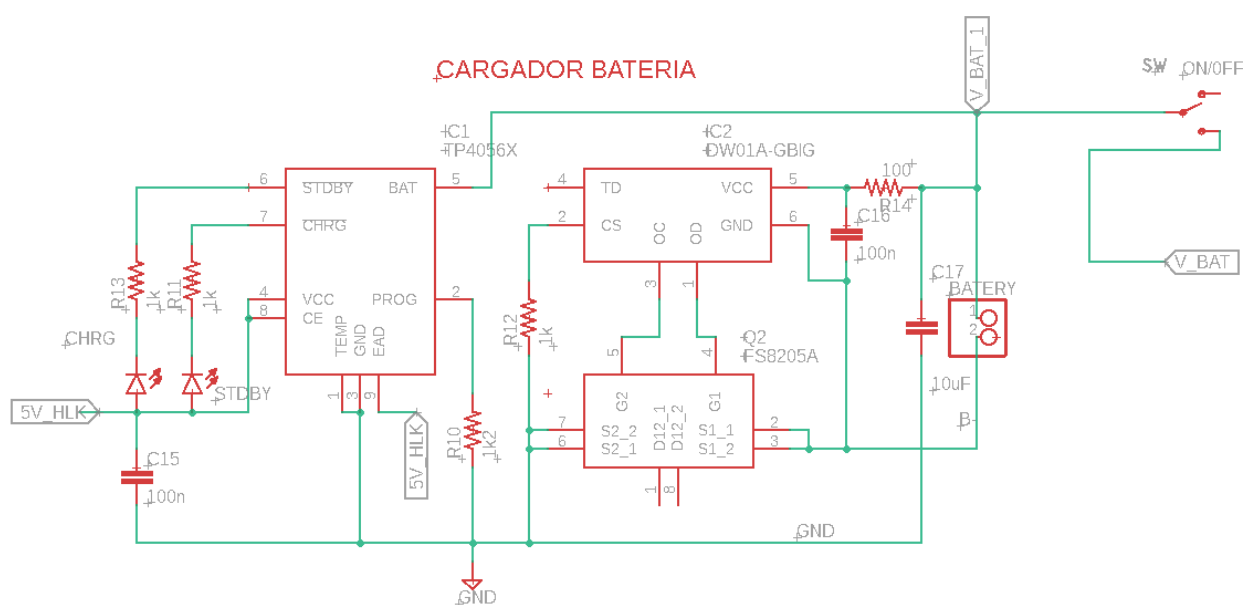


Figura 3.16: Esquema de implementación del circuito cargador de baterías y su etapa de protección

El circuito diseñado cuenta con dos indicadores luminosos la tabla 3.2 muestra el estado de los Leds para varios estados de carga:

Estado de la batería	CHRG	STDBY
Cargando	Luminoso	Apagado
Carga terminada	Apagado	Luminoso
Voltaje de carga bajo o temperatura de la batería muy baja o muy alta	Apagado	Apagado
Sin batería	Parpadeo	Parpadeo

Tabla 3.2: Estado de los indicadores leds en función de carga de la batería

Se aprecia en la parte superior derecha del esquemático mostrado en la figura 3.2 un interruptor deslizable denominado SW ON/OFF el cual realiza la función de encendido o apagado del sistema en general.

3.6.3. Baterías 18650

El uso de baterías de tecnología electroquímica basada en el ion litio es uno de los dispositivos de almacenamiento de energía más empleado, actualmente es una tecnología ampliamente usada, gracias a que presenta elevados valores de energía específica y potencia.

El empleo de la batería modelo 18650 provee energía al sistema embebido sin requerir una conexión a la corriente alterna. Esto aporta una versatilidad fundamental en situaciones de emergencia, cuando la red eléctrica no está disponible. El dispositivo que estamos desarrollando en este trabajo de tesis mantendrá su operatividad y capacidad de transmitir información incluso en ausencia de alimentación eléctrica convencional, siempre que cuente con una conexión a Internet.

Cabe destacar que la batería 18650, ilustrada en la figura 3.17, entra en funcionamiento automáticamente al desconectar el dispositivo de la red eléctrica. Este comportamiento es posible gracias al diseño electrónico detallado en el esquemático 3.16. Esta característica permite al usuario desconectar el dispositivo de la red eléctrica y transportar el sensor a ubicaciones estratégicas para la toma de mediciones, sin que ello afecte el funcionamiento integral del sistema. En este proceso, el sistema embebido no se reinicia ni se desactiva en ningún momento

La carga de la batería está diseñado para solo cargar la batería cuando está, alcance una descarga por debajo de $\frac{9}{10}$ de la capacidad de

la batería, entonces el sistema embebido podrá mantenerse conectado a la red eléctrica sin necesidad de preocuparse si la carga constante de batería podría afectar a la misma.



Figura 3.17: Batería 18650 usada en el sistema embebido

Las características principales de la batería usada son las siguientes:

Especificaciones técnicas	Descripción
Matrícula	18650
Tecnología empleada	Ion-litio
Capacidad	2.6 Ah
Tensión nominal	3.7 v
Dimensiones	18.4 mm x 65 mm, diámetro de 18 mm
Rango de temperatura de funcionamiento	Carga: 0 a 45° Descarga: -20°C a 60° C.
Tensión de carga	4.280 ± 0.05 v
Corriente de carga	Estándar: 1000 mA Rápida: 2600mA
Peso	4.7 gramos

Tabla 3.3: Especificaciones nominales de la batería 18650.

3.6.4. Convertidor elevador de voltaje

El circuito integrado para realizar la conversión de elevación de voltaje es el SDB6, con empaquetado SOT-23 de 6 pines.

Ideal para aplicaciones de baja potencia a una frecuencia de 1.2 MHz, el arranque interno de la conversión da como resultado una pequeña corriente de entrada y en consecuencia prolonga la vida útil de la batería. Tomando en cuenta las consideraciones del fabricante [60], se hace uso del circuito básico de aplicación.

Para nuestro caso es necesario tener en cuenta que la batería otorga máximo 4.2 volts, y el elevador de voltaje deberá proporcionar $\approx 5,0$

volts. ya que nuestro sistema embebido hace uso de dos voltaje 5.0 volts y 3.3 volts, por lo cual se hace la elección del primer valor del voltaje para posteriormente hacer la regulación a 3.3 volts.

El voltaje de referencia interno de nuestro convertidor elevador es típicamente de 0.6 volts, tomando en cuenta lo anterior, podemos realizar la configuración de salida del voltaje de nuestro convertidor, haciendo uso de la expresión matemática 3.1:

$$V_{Salida} = V_{Referencia} * \left(1 + \frac{R_{17}}{R_{18}}\right) \quad (3.1)$$

En lo cual la resistencia R_{17} y R_{18} observadas en la topología de la figura 3.18 se muestra con una configuración de divisor de voltaje donde la entrada la concede el pin numero 6 del circuito integrado SDB628, denominado FB el cual es la retroalimentación del convertidor, Donde los valores del voltaje de referencia es típicamente de 0.6 volts y el voltaje de salida debe ser ≈ 5.0 volts, se consideran los valores comerciales de $R_{17} = 5,1K \Omega$ Y $R_{18} = 680 \Omega$ el voltaje de salida da como resultado 5.1 volts, lo cual entra en el rango de tolerancia de alimentación de los componentes siguientes del sistema embebido.

$$5,1 \text{ volts} = 0,6 \text{ v} * \left(1 + \frac{5100 \Omega}{680 \Omega}\right) \quad (3.2)$$

La selección de los componentes activos como lo son el capacitor, así como el componente activo necesario para el funcionamiento que es un diodo schottky son seleccionados en función a la propuesta dada por el fabricante en el documento de especificaciones del SDB628 mostrados en la hoja de datos[60].

CONVERTIDOR ELEVADOR

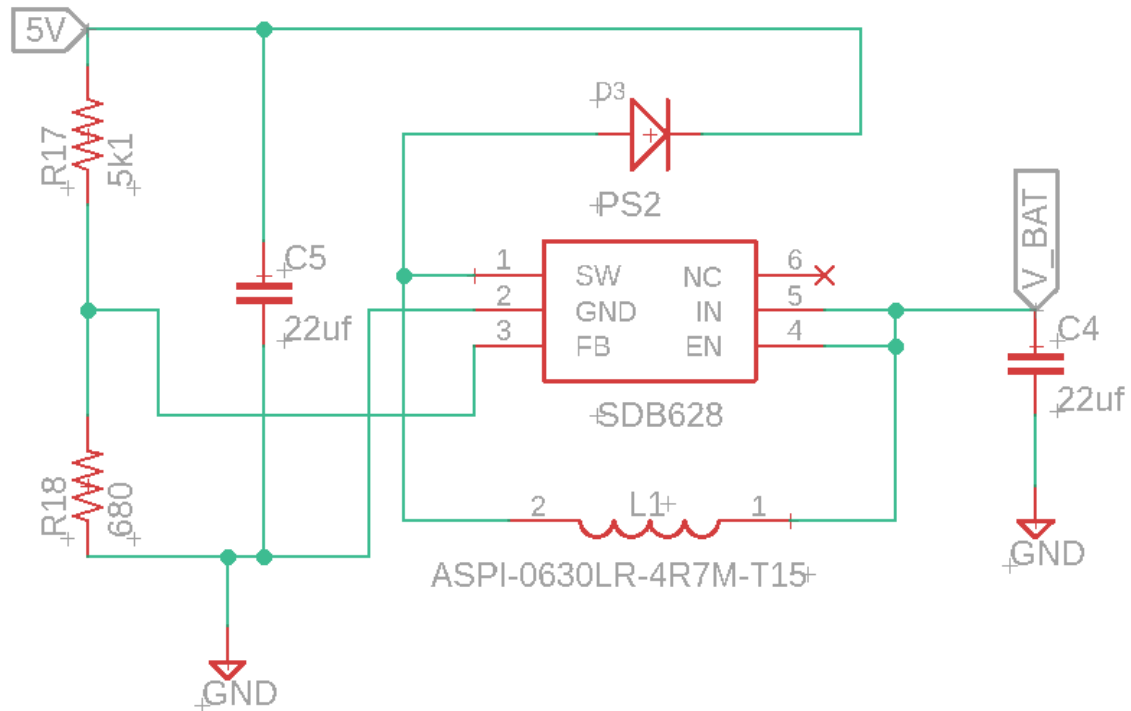


Figura 3.18: Esquema de implementación del módulo de conversión de AC-DC.

Para las consideraciones de diseño se toman en cuenta:

- Los condensadores de entrada y salida deben colocarse cercanos al circuito integrado SDB628 y ser conectados a un plano de tierra, con la finalidad de reducir el ruido de acoplamiento.
- La tierra conectada al circuito integrado debe estar en una área de plano de tierra considerablemente grande para la disipación de calor y la protección contra el ruido.
- Mantener las pistas de la PCB lo más cortas posibles.
- Evitar la cercanía de dispositivos con interferencias de ruido.

3.7. Implementación de la versión final del sistema embebido en placa de prototipo electrónico

Se realiza una prueba en placa de prototipo electrónicos con los componentes THT (*Through-Hole Technology*) usados en el diseño an-

terior propuesto en la figura 3.2, sin embargo se añaden unos módulos comerciales que contengan los componentes agregados vistos en la sección anterior, como lo es el módulo elevador de voltaje, convertidor de corriente alterna a corriente directa, la batería de tecnología LiPo y finalmente el sensor NDIR Senseair S8, la prueba hecha se observa en la figura 3.19.

Se observa un funcionamiento óptimo del sistema embebido. En las siguientes secciones se describirá el proceso de evaluación del circuito propuesto.

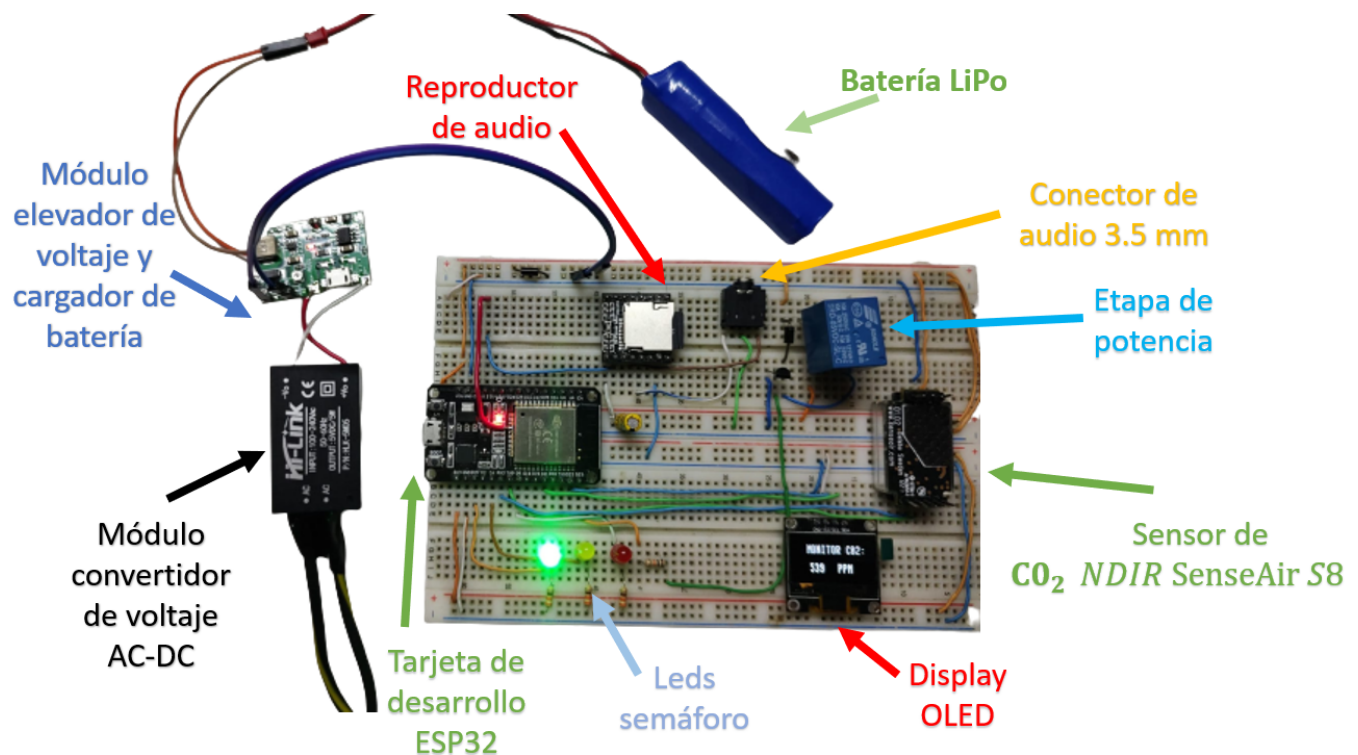


Figura 3.19: Prueba de la versión final del sistema embebido en placa de prototipos electrónicos.

3.8. Pruebas de funcionamiento del sistema embebido en placa de prototipos electrónicos, implementando un sensor de tecnología NDIR

Anteriormente en la sección 2.2.2 o se había hablado del sensor a utilizar en nuestro diseño electrónico, se describieron los parámetros y especificaciones del sensor con principio de funcionamiento con tecnología

NDIR para la detección de PPM (Partículas por millón) de CO_2 .

Recapitulando la información anterior acerca del sensor de CO_2 en esta sección se centra en detallar la metodología para el uso del sensor, para el uso del sensor Sense Air s8 es necesario el uso de dos pines de alimentación uno de 5 volts y el otro tierra. Se hace uso de un protocolo de comunicación serial donde existe un pin de transmisión y otro de recepción, finalmente se tiene un pin de calibración.

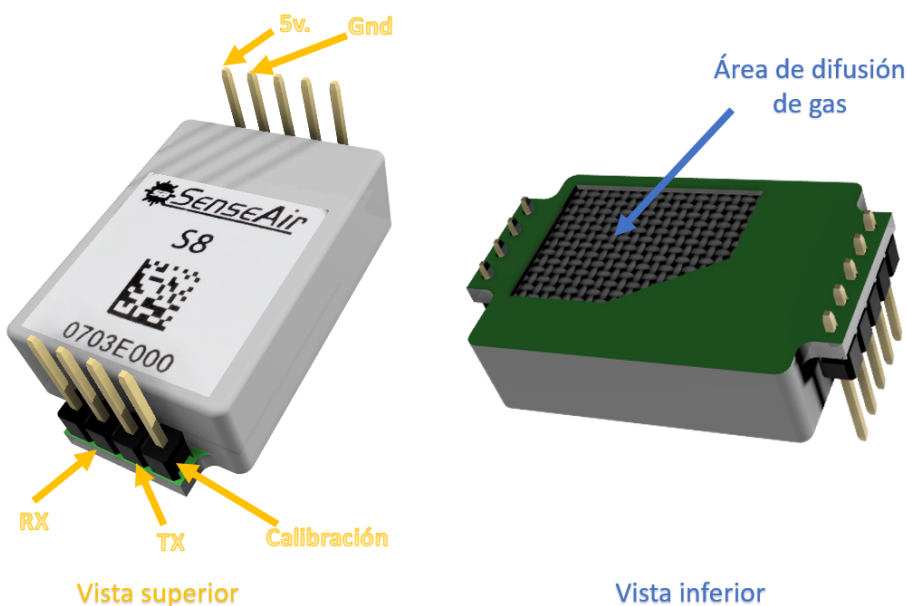


Figura 3.20: Sensor NDIR SenseAir S8.

Se puede observar la asignación de pines y el área de difusión del sensor en la figura 3.20.

El área de difusión de gas es posicionado de tal modo que el gas ingrese al sensor de manera eficiente, sin algún dispositivo que se encuentre de manera paralela a la rejilla de ingreso del gas, haciendo uso de las especificaciones mencionadas por el fabricante [34].

La calibración general se realiza manualmente utilizada antes de hacer uso del protocolo de comunicación se realiza puentando el pin de calibración con la tierra en general del sistema, en un lapso de 4 a 8 segundos, esto se realiza en un espacio donde se pretende exista la menor condición de contaminación al aire, donde idealmente se debería presentar un valor de 400 PPM de CO_2 .

La calibración mencionada con anterioridad es de vital importancia para poder iniciar con el protocolo de comunicación del sensor, a partir

de la calibración manual el SenseAir s8 no requiere de mantenimiento en entornos residenciales ya que el fabricante realiza una implementación de algoritmo de autocorrección incorporado, denominado ABC (*Automatic baseline*). Esto es una función para lograr que el sensor de gas no necesite mantenimiento por un largo periodo, teniendo una vida útil de al menos 15 años y no requieran calibración adicional cuando se utiliza en aplicaciones normales de aire interior [34].

3.8.1. Protocolo de comunicación sensor SenseAir S8

El sensor para la captación de CO₂ mediante un protocolo de comunicación serial MODBUS, el sensor es un esclavo y nunca iniciará la comunicación. El sistema maestro en este caso el módulo electrónico ESP-32 inicia la transacción para leer los valores de C02 del registro correspondiente. El sistema host también verificará el estado del sensor periódicamente para determinar si se está ejecutando sin detectar fallas [61].

Acorde a las especificaciones de la comunicación Modbus, el tamaño de paquete son máximo máximo 255 bytes incluyendo la dirección de la memoria. El sensor provee una comunicación con niveles lógicos seriales CMOS de transmisión (TX) y de recepción (RX). Las líneas de comunicación se alimentan directamente al microcontrolador del sensor. Para iniciar la habilitación del sensor y el envío de datos se conecta el módulo ESP-32 hacia el sensor, con el siguiente esquema mostrado en la figura 3.21.

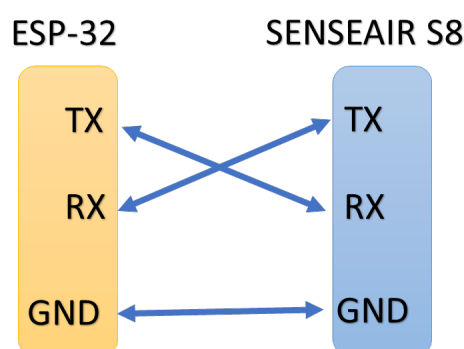


Figura 3.21: Comunicación serial entre el Módulo ESP-32 y el SenseAir S8.

Para realizar la comunicación serial se realiza la configuración del protocolo con las especificaciones mostradas en la tabla 3.4.

Especificaciones técnicas	Descripción
Velocidad de transmisión	9600 Bits/segundo
Número de bits de transmisión	8
Paridad	No
Voltaje lógico de transmisión	3.3 v
Bits de paro	1

Tabla 3.4: Especificaciones usadas para la comunicación con el sensor Senseair S8.

Se inicializa la configuración del sensor, enviando una cadena con una longitud de 7 bits donde se envían los caracteres hexadecimales mostrados en la tabla 3.5, donde tenemos un buffer el cual se llena de igual manera de 7 bits en hexadecimal, lo que cambia son los bits en la posición 3 y 4, se denominan bit *HIGH* y bit *LOW* respectivamente. Posteriormente en cada medición realizada se deja un retardo de 2 segundos.

Posición del bit	0	1	2	3	4	5	7
Hexadecimal enviado	0xFE	0x44	0x00	0x08	0x02	0xF9	0x25
Hexadecimal recibido	0xFE	0x44	0x00	<i>H</i>	<i>L</i>	0xF9	0x25

Tabla 3.5: Cadena de Bits enviados y recibidos por el sensor Senseair S8.

Posteriormente se hace uso de los bits *HIGH* y *LOW* y mediante la expresión matemática la cual se muestra en la ecuación 3.3 se proporciona el valor adquirido por el sensor, el cual son las PPM del CO₂, la cual es proporcionada por el fabricante [61].

$$CO_2 = (256 * \text{Byte high}) + \text{Byte low} \quad (3.3)$$

Un ejemplo de lectura del sensor de CO₂ puede verse en la expresión 3.4, donde se tiene que *Bit high* = 0x02 y el *Bit low* = 0xEB

$$CO_2 = (256 * 2) + 235 = 747 \text{ PPM} \quad (3.4)$$

Se presenta en la figura 3.22 un algoritmo de la metodología empleada para el uso del sensor de CO₂.

El algoritmo empleado para las pruebas de transmisión de datos del sensor Senseair S8 hacia el ESP-32, en la sección 6.3.1.

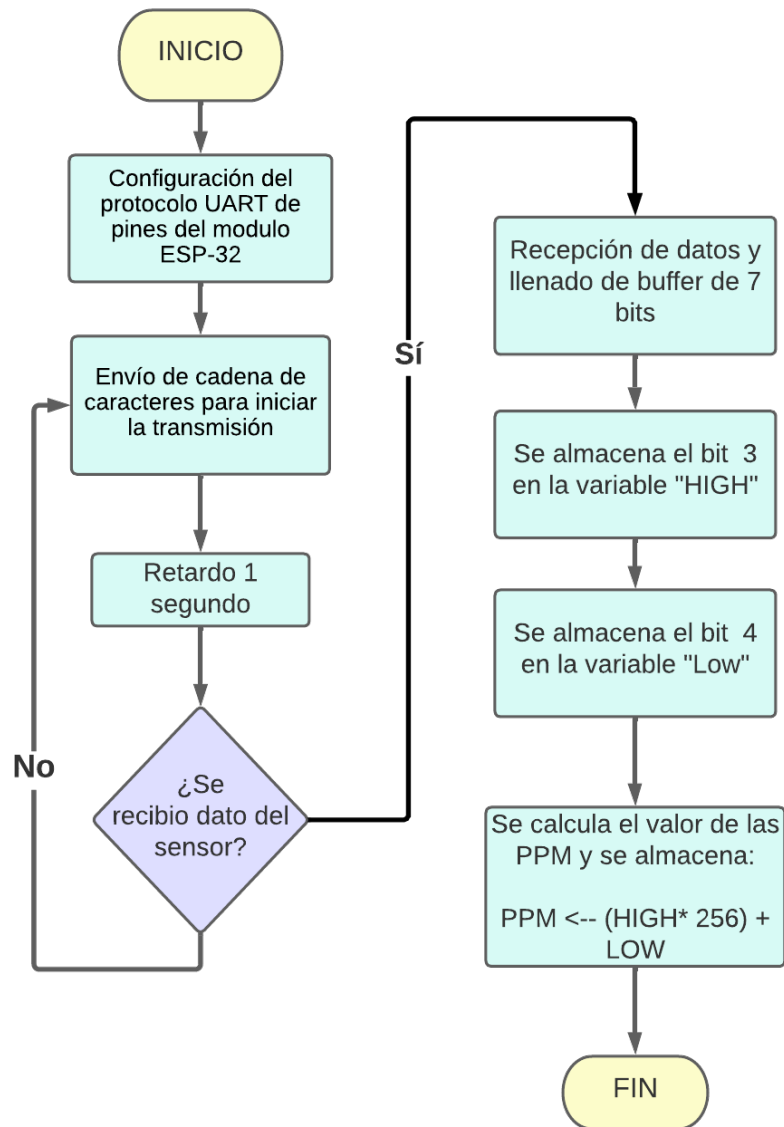


Figura 3.22: Comunicación serial entre el Módulo ESP-32 y el SenseAir S8.

Se presenta una explicación de cada byte en la cadena de hexadecimales usada para la comunicación con el sensor SenseAir s8, que se presenta en la tabla 3.5:

- FE: Este es el byte de inicio de la trama. Todos los comandos al sensor comienzan con este byte.

- 44: Este es el comando para solicitar una medición del sensor. Le comunica al sensor que está solicitando un valor de medición.
- 00: Este byte es el byte de datos de alta orden del comando. Especifica el número de bytes de datos que se enviarán a continuación. En este caso, estamos enviando 0 bytes adicionales de datos, por lo que este byte es 0.
- 08: Este byte es el byte de datos de baja orden del comando. Especifica el número de bytes de datos que se enviarán a continuación. En este caso, estamos enviando 8 bytes adicionales de datos, por lo que este byte es 8.
- 02: Este es el número de bytes de datos que se solicitan. En este caso, estamos solicitando 2 bytes de datos para la medición del CO₂.
- 9F: Este es un byte de verificación de suma. Es la suma de verificación de los bytes de la trama desde el byte de inicio hasta el byte anterior a este. Es utilizado por el sensor para verificar que la trama se recibió correctamente.
- 25: Este es el byte de fin de la trama. Todos los comandos al sensor terminan con este byte.

3.9. Diseño de PCB del sistema embebido

Se mostró en las secciones anteriores, el *software* usado para el diseño de la primer placa electrónica de prueba, se desarrolla una segunda versión escalando el diseño inicial con los módulos descritos anteriormente y el implemento de una batería y su carga. En esta sección se describe el desarrollo del diseño de PCB final, así como el diseño ECAD y los resultados de la implementación del sistema embebido.

Se muestra en la figura 3.23 el diseño de la placa electrónica, realizada en el *software Eagle*, se presenta una visualización translúcida de la tarjeta donde se aprecia dos capas de conexiones, con color rojo tenemos la capa superior y en color azul la capa inferior.

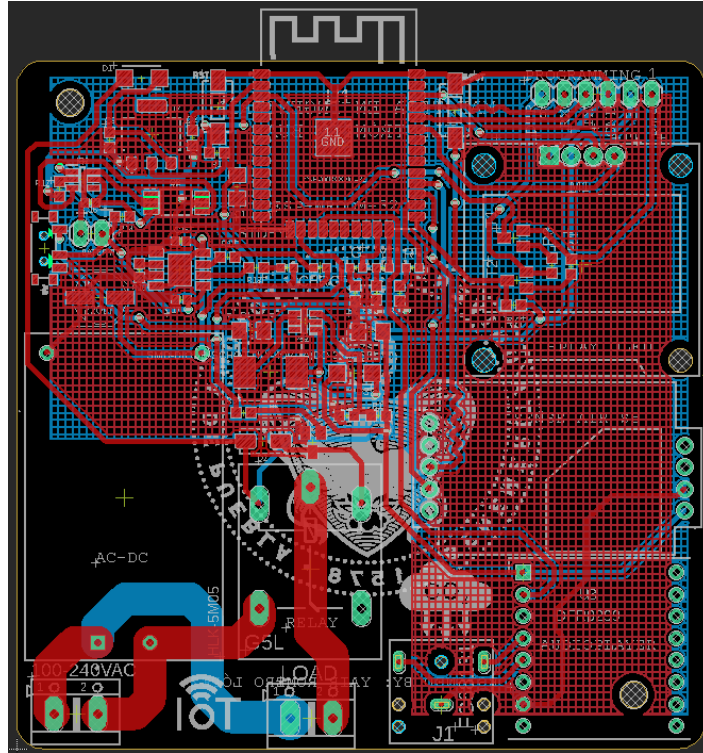


Figura 3.23: Versión final de la PCB del sistema embebido.

Se hace uso de las reglas de diseño consideradas en la sección 3.1.2, añadiendo los módulos propuestos en la sección 3.6, sin embargo al ser un diseño más robusto se añaden ciertas consideraciones para el diseño electrónico, las cuales serán beneficiosas al realizar el montaje y el soldado de los componentes:

- Es necesario planificar la colocación de los componentes tomando en cuenta su funcionalidad, si existen conectores deben estar a los bordes de la tarjeta.
- Asegurar de que los componentes similares estén orientados en la misma dirección, ya que esto permitirá lograr una soldadura eficiente y sin errores.
- Colocar todos los componentes de montaje superficial (SMD) en la capa inferior para facilitar la soldadura de los mismos y permitir la detección de fallas de manera eficiente.
- Realizar un plano de tierra geométrico en ambas capas.
- Las conexiones de enrutamiento son cortas y buscando ser lo más directas posible entre los componentes.

- Hacer uso de una calculadora de ancho de pistas ya que se hace uso de señales digitales y señales de potencia, las cuales difieren de su tamaño [62].
- Identificar los componentes con mayor susceptibilidad a un calentamiento, y se crea un pad térmico haciendo uso de las referencias de diseño proporcionadas por el fabricante del dispositivo.
- Al final del proceso de diseño, se hace una serie de revisiones, teniendo en cuenta el proceso de control de calidad, se verifica las Reglas Eléctricas (ERC *Electric Rule Check*) y la Verificación de Reglas de Diseño (DRC *Design Rule Check*) para asegurar que se ha cumplido con todas las restricciones establecidas. Ambos sistemas permiten cumplir fácilmente con los anchos de espacio, los anchos de pistas, las configuraciones comunes de fabricación, los requisitos de alta velocidad y los cortocircuitos.

Para reducir los efectos de la EMI (Interferencia Electromagnética, por sus siglas en inglés)y garantizar que los dispositivos cumplan con los estándares de emisiones electromagnéticas, se deben seguir ciertas reglas de diseño. A continuación, se presentan algunas de las reglas de diseño EMI que fueron realizadas:

- Enrutamiento adecuado de señales: El diseño de la PCB (placa de circuito impreso) debe garantizar que las señales digitales y analógicas estén enrutadas de manera separada y cercana a la fuente de energía correspondiente. Esto ayuda a prevenir la interferencia cruzada entre las señales y reduce la susceptibilidad a la EMI.
- Planificación de la capa de tierra: Una buena planificación de la capa de tierra en la PCB es esencial para reducir la EMI. Se deben establecer planos de tierra sólidos y continuos que actúen como una referencia de voltaje estable y ayuden a eliminar los bucles de corriente no deseados.
- Uso de técnicas de filtrado: Se deben incorporar componentes de filtrado, como inductores, condensadores y ferritas, en el diseño para atenuar las interferencias electromagnéticas. Estos componentes pueden ayudar a eliminar ruidos no deseados en las líneas de alimentación y señal.
- Protección de cables: Los cables de interconexión deben ser enruta-

dos y protegidos adecuadamente para evitar que actúen como antenas y radien energía electromagnética. Se pueden utilizar trenzas, blindajes y filtros en los cables para reducir la EMI.

- Diseño de carcasa adecuado: La carcasa o el recinto del dispositivo deben ser diseñados de manera adecuada para evitar la radiación de la energía electromagnética al entorno. Se deben utilizar materiales conductores o apantallamientos para bloquear las emisiones y mejorar la inmunidad del dispositivo a la EMI externa.
- Gestión térmica adecuada: El sobrecalentamiento de los componentes electrónicos puede afectar negativamente el rendimiento y la emisión electromagnética de un dispositivo. Un adecuado diseño de gestión térmica, con disipadores de calor y ventilación adecuados, ayuda a mantener la temperatura controlada y minimiza los efectos de la EMI.

Se integran 13 etapas a la versión final de la placa electrónica las cuales se pueden apreciar en el esquemático de la figura 6.2, a continuación se en lista dichas etapas enumeradas iniciando de la esquina superior izquierda a derecha del esquema eléctrico, y se señalan en la figura 3.24 la ubicación de las mismas.

- 1.- Conversión AC-DC.
- 2.- Cargador de batería batería.
- 3.- Convertidor elevador de voltaje.
- 4.- Regulador de voltaje 3.3 v.
- 5.- ESP-32.
- 6.- Leds semáforo
- 7.- Transistores de auto-programación.
- 8.- Pines para la programación de la tarjeta.
- 9.- Botones de boot y habilitación.
- 10.- Reproducción de audios
- 11.- Sensor de CO₂ con tecnología NDIR.
- 12.- Display OLED
- 13.- Etapa de potencia.

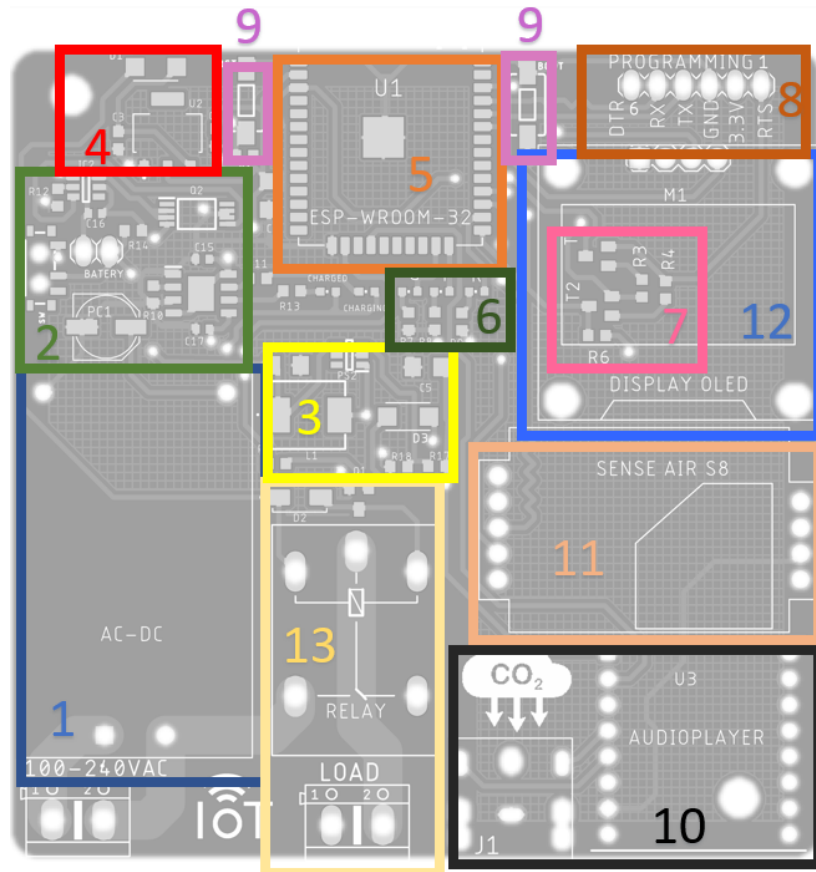


Figura 3.24: Visualización de la PCB y sus respectivos bloques.

3.10. Diseño ECAD para la implementación de la estructura externa de protección del sistema embebido

La protección de los sistemas electrónicos es de suma importancia para implementar el sistema embebido diseñado en este proyecto, el uso de materiales aislantes para la protección de los dispositivos y las líneas de alimentación son importantes para realizar pruebas.

Es por ello que se hace uso de un *software* dedicado a la implementación de diseños 3D el cual se describió en la sección 2.1.2 y 3.10, se inicia con el diseño de los componentes de la PCB en 3D, permitiéndonos la correcta colocación y la verificación de tamaños de los componentes.

Se ajusta cada componente en función de los *pads* o los conectores correspondientes y finalmente se realiza una visualización 3D, la cual se podrá observar en la siguiente dirección web: https://drive.google.com/file/d/1EPXv8jh9XewpWBjiHKRtH_9P0s5aymeC/view?usp=sharing.

Se realiza una renderización de la PCB con los ensamblajes de los dispositivos electrónicos, la cual se puede observar en la figura 3.25.

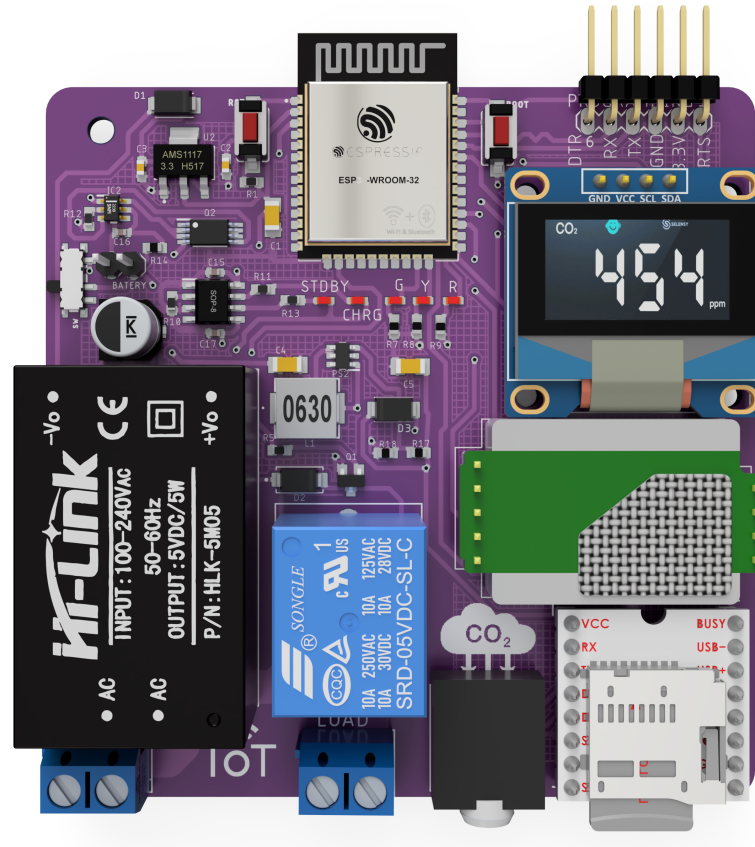


Figura 3.25: Renderizado de la versión final de la PCB.

Se procede a realizar un diseño 3D para una carcasa la cual proteja a nuestro sistema electrónico y las líneas de alimentación, se realiza un diseño mecánico considerando como principal material el ácido poliláctico (PLA), debido a su facilidad de impresión, es un material rígido y resistente que se utiliza comúnmente para piezas mecánicas y estructurales, lo que lo hace ideal para prototipos mecánicos que se puedan construir en poco tiempo, El PLA se puede lijar, pintar y perforar fácilmente, lo que lo hace perfecto para el prototipado de este sistema ya que se puede requerir ajustes y modificaciones.

El diseño propuesto como producto final, esta constituido por la segmentación de 5 bloques, los cuales se pueden observar en la figura 3.26, se observa a la derecha el ensamble final y a la izquierda la segmentación de los bloques mecánicos, los cuales se enumeran a continuación:

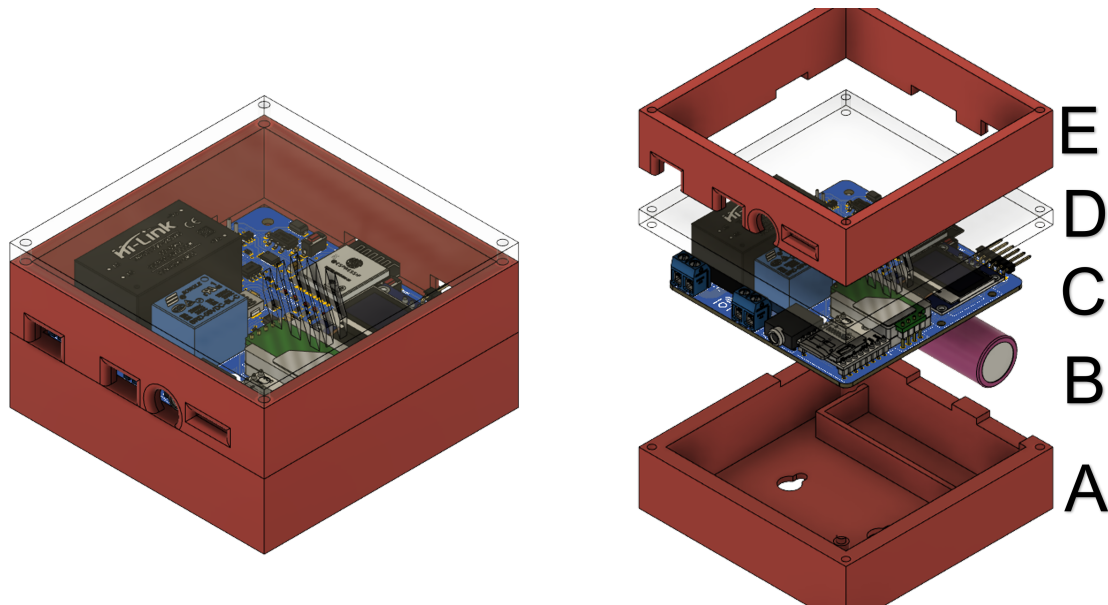


Figura 3.26: Renderizado del producto final del sistema embebido.

- A: Base de PLA de 91 mm de largo, 91 mm de ancho y 23 mm de altura, el cual contiene orificios en la base para la sujeción del dispositivo en una superficie con tornillos, se encuentra un espacio que delimita la batería y asegura la posición de la misma.
- B: Se procede a realizar el modelo 3D de la batería 18650, la cual es colocada en el segmento A.
- C: Modelo 3D de la PCB del sistema embebido.
- D: Cubierta del sistema mecánico, el cual es traslucida al ser acrílico con un grosor de 3 mm, la cual hará posible la visualización del sistema electrónico, las notificaciones luminosas de los leds y la pantalla, es importante notar que esta cubierta cuenta con una rejilla paralela al sensor de CO_2 la cual hará posible el ingreso de dicho gas en el área de difusión del sensor, descrita en la figura 3.20.
- E: Base intermedia de PLA de 91 mm de largo, 91 mm de ancho y 22 mm de altura, la cual tiene la funcionalidad de ser el soporte de la cubierta de acrílico y tener orificios necesarios para los conectores de alimentación, audio, programación del dispositivo y una abertura de la antena *Wi-Fi* del sistema embebido.

Se realiza una simulación del producto final a través de un render, haciendo uso de las secciones descritas anteriormente, se muestra

el diseño del sistema embebido y el empalme de la carcasa. El sistema embebido es compacto, lo que lo hace ideal para su uso en entornos de uso de la vida cotidiana, los cuales no exigen una alta exigencia industrial. La carcasa ha sido diseñada para ser resistente y duradera, capaz de proteger el sistema embebido de los factores ambientales y los golpes que puedan producirse en el lugar de su uso. La combinación del sistema embebido y la carcasa garantiza la seguridad y el rendimiento del sistema.

Se muestra la simulación del producto final en la siguiente dirección web: <https://goo.su/zNYkje>.

3.11. Adición de sensores HVAC

Para garantizar el bienestar de los ocupantes de los edificios, es necesario que el ambiente interior cumpla con los estándares de calidad del aire interior, así como con los niveles adecuados de confort térmico. Dicho bienestar depende de diversos factores, como la temperatura y velocidad del aire, la humedad, la disposición de la ventilación, la ropa utilizada, y la presencia de contaminantes tanto gaseosos como en partículas [63].

Los edificios pueden presentar una gran variedad de contaminantes cuyas concentraciones pueden sobrepasar umbrales críticos debido a las actividades humanas, los materiales de construcción utilizados, así como a la calefacción y cocción de alimentos, entre algunas otras actividades cotidianas[64].

Los sensores *HVAC* son dispositivos que se utilizan en los sistemas de calefacción, ventilación y aire acondicionado (por sus siglas en inglés) para monitorear y controlar el entorno interior de un edificio. Estos sensores están diseñados para medir diversos parámetros ambientales, como la temperatura, la humedad, la calidad del aire y el flujo de aire. Son esenciales para garantizar un ambiente interior cómodo y saludable, así como para mejorar la eficiencia energética. Por ejemplo, los sensores de temperatura y humedad pueden ajustar automáticamente la salida de aire acondicionado o calefacción para mantener una temperatura y humedad óptimas. Los sensores de calidad del aire pueden detectar la presencia de contaminantes y ajustar la ventilación para mejorar la calidad del aire interior [13].

Los resultados obtenidos en la aplicación de detección de altos niveles de CO₂ y un funcionamiento óptimo del sistema embebido, muestran una escalabilidad del sistema, y de esta manera, aumentar su potencial para ser utilizado en otras aplicaciones, como los sistemas *HVAC*, es por esta razón que esta sección que se añaden tres nuevas variables a nuestro sistema embebido: **Temperatura, humedad relativa y presión atmosférica** .

Para llevar a cabo una vigilancia efectiva de contaminantes en ambientes interiores es fundamental elegir los sensores adecuados. Existen estudios en la literatura, los cuales muestran que la IAQ se ve muy afectada por varios parámetros de confort térmico, especialmente la temperatura y la humedad [65]. Es por ello que se hace una selección del sensor el cual cumpla con un funcionamiento satisfactorio de las mediciones de las variables mencionadas. Se observa una recopilación de estudios los cuales mencionan el uso del sensor BME280 para medir parámetros de confort térmico [13]. La principal ventaja de BME280 es que también puede medir la presión atmosférica junto con la temperatura y la humedad. temperatura y la humedad.

3.11.1. Sensor BME280

El sensor BME280 el cual se aprecia en la figura 3.27, es un sensor de temperatura, la humedad relativa del aire y presión atmosférica, fabricado por la compañía Bosch Sensortec. El sensor utiliza una tecnología de compensación de temperatura y presión para mejorar la precisión de las mediciones, lo que significa que es capaz de proporcionar resultados precisos incluso en condiciones climáticas cambiantes.

El rango de medición de la presión atmosférica del sensor BME280 va desde 300 a 1100 hPa, con una precisión de ± 1 hPa. Para medir la temperatura, el rango de medición es de -40°C a $+85^{\circ}\text{C}$, con una precisión de $\pm 1^{\circ}\text{C}$. En cuanto a la medición de la humedad relativa, el rango de medición es del 0 % al 100 %, con una precisión de ± 3 %.

El sensor BME280 3.27 es muy fácil de integrar en sistemas embebidos y cuenta con una interfaz digital I²C o SPI de alta velocidad. Además, tiene un bajo consumo de energía, con un consumo de corriente en espera de solo 0,1 microamperios. Debido a sus características, el sensor BME280 es ideal para su uso en aplicaciones como el control del

clima, la monitorización ambiental y la monitorización de la calidad del aire.

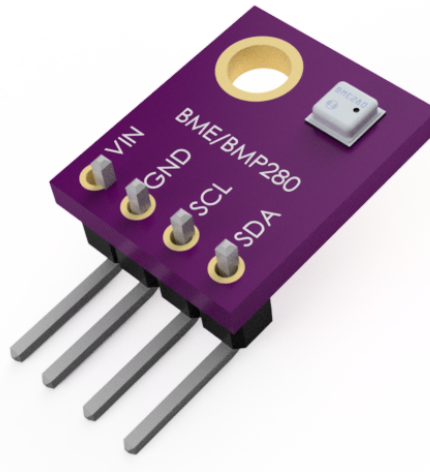


Figura 3.27: Renderizado del sensor BME280.

El sensor muestra la característica de realizar mediciones de tres parámetros por lo cual es una opción recomendable. Al ser un sistema embebido escalable y versátil la adición del sensor se realiza de manera sencilla.

Se establece la comunicación I²C al SoC ESP-32 y se elige una dirección dada por el fabricante la cual es 0x76 o 0x77, y su alimentación eléctrica es de 3.3 volts.

Es importante puntualizar que el sensor se coloca en una PCB exterior al sistema embebido, esto con la finalidad de darle mayor fiabilidad a las mediciones realizadas y sea mínima la afectación de temperatura generada por el mismo sistema embebido. En la figura 3.35 se muestra la colocación del sensor ya en el sistema embebido ensamblado y ajustado a su respectiva estructura.

La biblioteca implementada en el *software Microphyton* se adjunta en la sección 6.3.2.



Figura 3.28: Colocación del sensor.

3.12. Prototipo final

Para la creación de la estructura se hace uso de la impresión 3D en una impresora de la marca Ender 3 pro. El proceso se realiza solo de los segmentos A y E. Se divide el modelo en capas finas para preparar el modelo para la impresión, y se realiza la carga del archivo STL (acrónimo de *Standard Tessellation Language*) este es un formato de archivo utilizado en impresión 3D y en CAD. Este formato de archivo representa la geometría de un objeto tridimensional mediante una malla de triángulos planos conectados entre sí, que se utilizan para definir la superficie externa del objeto. El archivo STL contiene información sobre la posición de los vértices de los triángulos, la orientación y la normal de la superficie de los triángulos, y otros datos necesarios para la creación de un objeto 3D a través de la impresión [66], este archivo se visualiza en el *software Ultimate Cura*, para colocar la pieza de manera correcta en la cama de la impresora 3D, como se muestra en la figura 3.29.

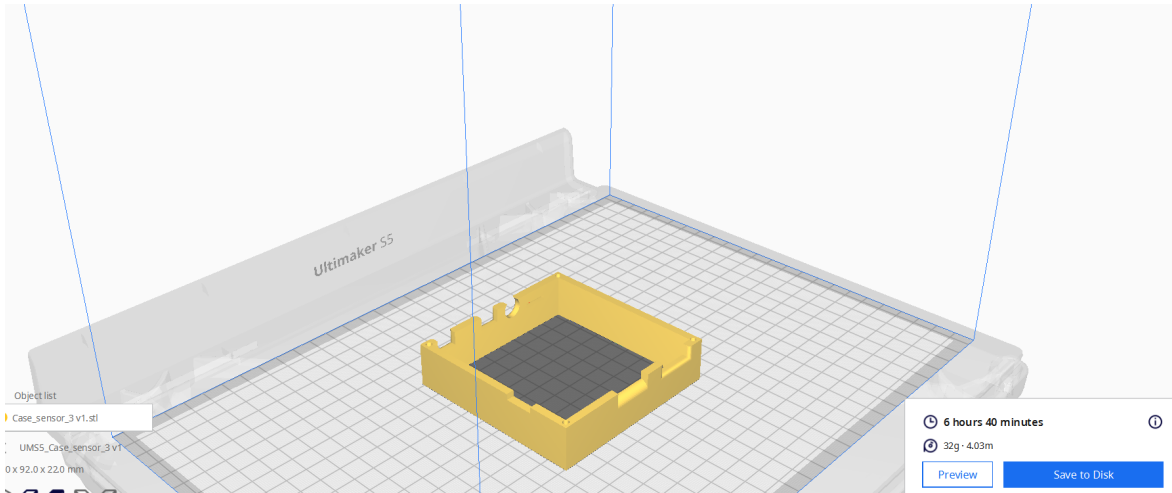


Figura 3.29: Visualización de los archivos STL en el *software Ultimate Cura*.

Posteriormente, se carga filamento de material en este caso el PLA. La impresora calienta y derrite el filamento para que se pueda extruir en una fina capa que se solidifica rápidamente al enfriarse. Este proceso se repite capa por capa hasta que se completa el objeto tridimensional. El resultado final es una réplica exacta del modelo de diseño en 3D, lista para su uso, el producto final se visualiza en la figura 3.30



Figura 3.30: Producto final del sistema embebido.

3.13. Implementación y recolección de datos en el *software* IoT en *Thingspeak*

Se visualiza en la figura 3.31 los datos recibidos por el *software* de *Thingspeak*, como se plantea en la sección 3.11 la adición de nuevas variables esta presente en la gráfica de los valores obtenidos, de igual manera la exportación de dichos datos es posible en un archivo con formato .CSV, con un límite de recepción de datos de tres millones.

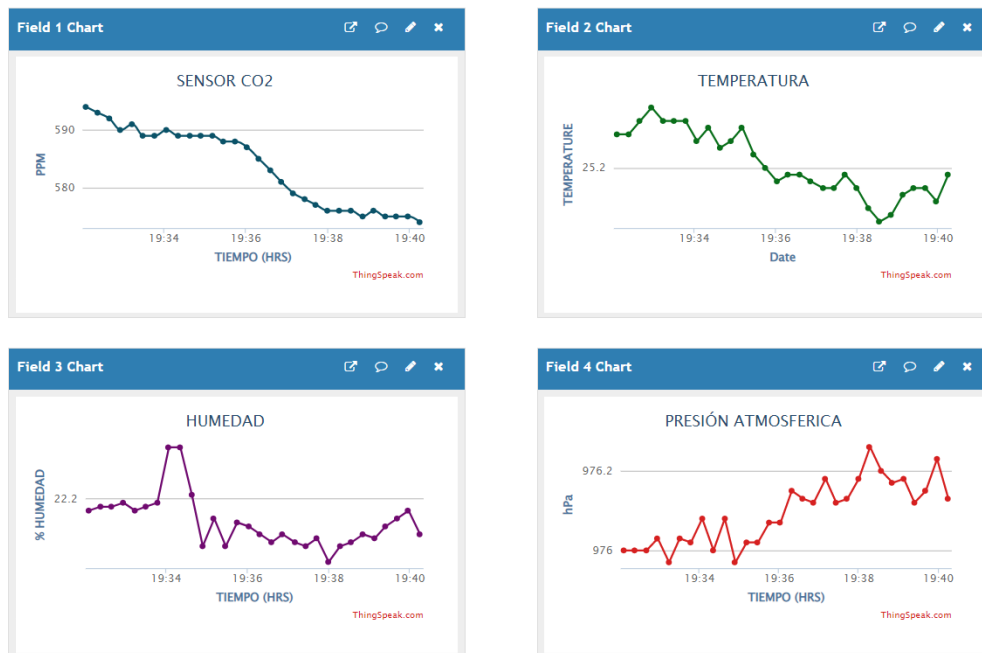


Figura 3.31: Visualización de las variables medidas en el *software* *Thingspeak*.

El *software* *Thingspeak* ofrece configuración sobre las visualizaciones de los canales y las gráficas generadas, tales como el número de datos mostrados. El canal en el cual se muestran los datos, es de orden público y se puede acceder a los datos, solo con el ingreso de la plataforma <https://thingspeak.com> y con el número de identificación del canal, en este caso es el **1863281**.

Es posible realizar una exportación de datos como se muestra la tabla 3.6, se visualizan que cuatro campos en función de las variables monitoreadas, es posible descargar los archivos por campos o los cuatro campos en un mismo documento, es importante señalar que existen tres diversos formatos para la exportación de los datos.

Export recent data	
CO2_TEMP_HUMIDITY_PRESSURE Channel Feed:	JSON XML CSV
Field 1 Data: co2	JSON XML CSV
Field 2 Data: TEMPERATURE	JSON XML CSV
Field 3 Data: HUMIDITY	JSON XML CSV
Field 4 Data: PRESSURE	JSON XML CSV

Tabla 3.6: Opciones para exportar los datos almacenados en *ThingSpeak*.

3.14. Implementación recolección de datos en una aplicación para *smartphone* con sistema operativo Android

La visualización de los datos se lleva acabo en una aplicación móvil Android, denominada *ThingView* que ofrece una experiencia visualización sin complicaciones de los diversos campos del canal. Con solo ingresar el ID del canal, Es posible acceder a una visión clara y detallada de los datos. Si se trata de un canal público, la aplicación respetará la configuración del sistema operativo, incluyendo aspectos como el color, la escala de tiempo, el tipo de gráfico y el número de resultados. Los gráficos de líneas y columnas son compatibles con la versión actual, mientras que las tablas de spline se presentan en forma de gráficos de líneas. Por otro lado, en el caso de los canales privados, la configuración predeterminada se utilizará para mostrar los datos, dado que la clave API no permite el acceso a las configuraciones de ventanas privadas.

Se presenta en la figura 3.32 la visualización de los últimos cien datos, y se presentan el dato mas reciente recibido con la fecha y hora, la medición mínima y máxima. La aplicación tiene un auto ajuste de los datos e igualmente actualización automática y manual, la aplicación en formato .apk es posible descargarla en el siguiente enlace: https://play.google.com/store/apps/details?id=com.cinetica_tech.thingview&hl=es_419&gl=US.

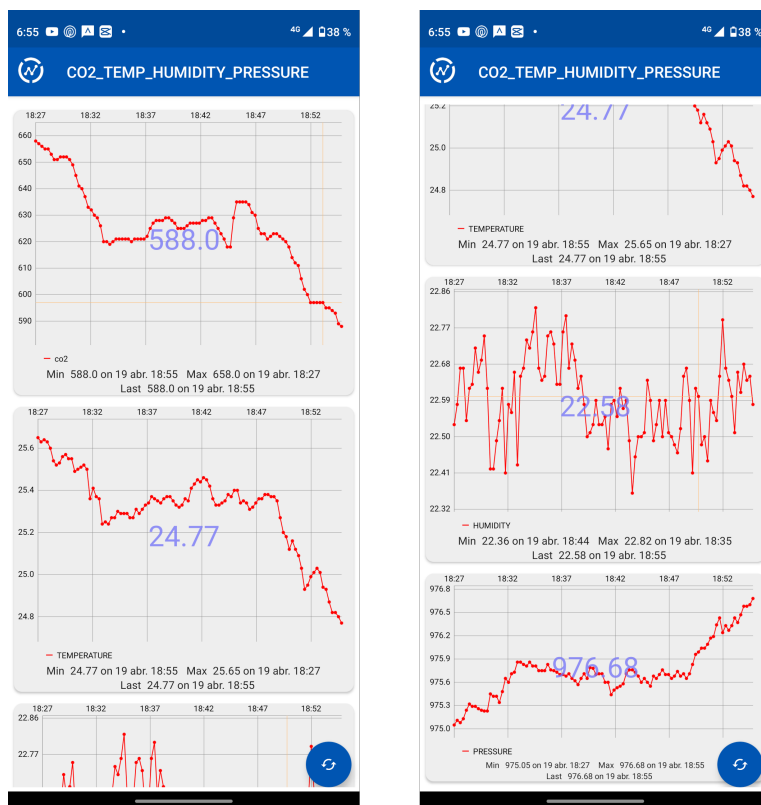


Figura 3.32: Presentación de la *dashboard* visualizada en el aplicación móvil.

3.15. Generador de tweets por el sistema embebido

Se añade una funcionalidad en la cual el sistema embebido haciendo uso de la aplicación *ThingTweet* es capaz de vincularse a una cuenta de *Twitter*, de esta manera el dispositivo es capaz enviar alertas a través de la misma cuenta. En este caso se genera una notificación cuando las condiciones de riesgo para la convivencia, no son propicias y existe un riesgo de contagio de COVID-19, esto se basa a la condicional si el $\text{CO}_2 > 1000$ PPM se publica un mensaje en la cuenta denominada Sensor CO_2 informando sobre las mediciones realizadas del sensor de CO_2 , Temperatura, humedad relativa y presión atmosférica. Se muestra un ejemplo de lo dicho anteriormente en la figura 3.33.



Figura 3.33: Visualización del *Tweet* generado por si el sistema embebido al alcanzar un nivel de $CO_2 > 1000$ ppm.

Para dicha funcionalidad es necesario hacer la vinculación de la cuenta de *ThingTweet*, con la cuenta de *Twitter* ingresando las credenciales correspondientes del usuario. Posteriormente mediante una Api que se genera y se coloca en el código de *MicroPhyton*, se envía los datos con el protocolo descrito en la sección 2.5.1.

Se realiza una función en *MicroPhyton* para la conexión del sistema embebido a nuestra cuenta de *Twitter*, dicho algoritmo se muestra en la sección de algoritmos 6.3.3.

Finalmente con la integración de lo planteado en la sección 3.11.1 y 3.15, se realiza un último ajuste al algoritmo en *MicroPhyton* para la realización del *firmware* implementado en el sistema embebido, se presenta dicho algoritmo en la figura 3.34.

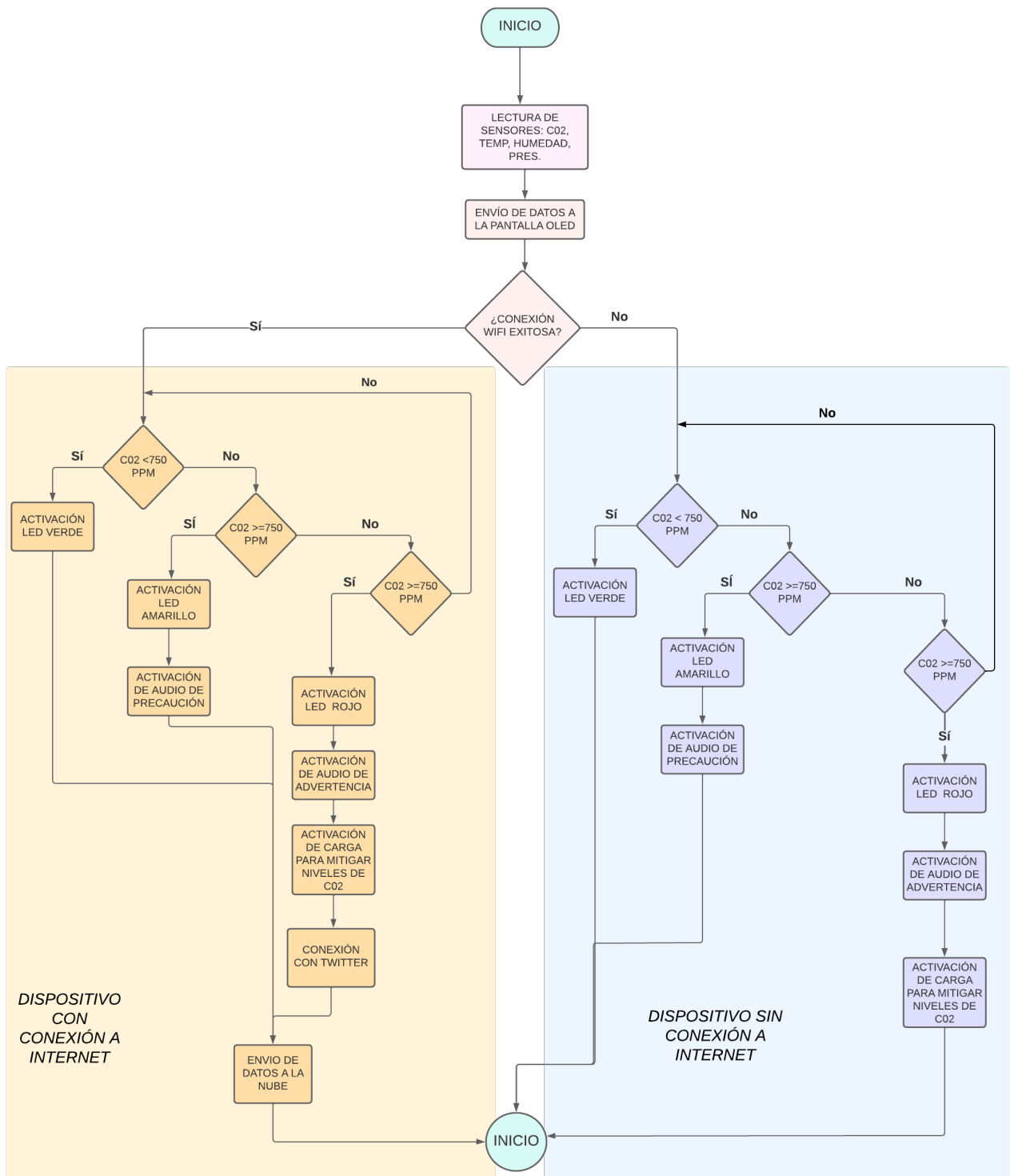


Figura 3.34: Algoritmo final del *firmware* implementado en el sistema embebido.

El algoritmo implementado en el sistema embebido está diseñado para funcionar sin necesidad de conexión a internet. En la Figura 3.35 se pueden observar las dos imágenes que se muestran en la pantalla OLED.

1. La primera imagen muestra la medición de CO₂ y el estado actual de conexión a internet del sistema.
2. La segunda imagen muestra la medición de temperatura en grados centígrados, la humedad en porcentaje y la presión atmosférica en hectopascales.



Figura 3.35: Imágenes mostradas en la pantalla del sistema embebido.

3.16. Aplicación del sistema embebido

Se realizan pruebas de monitoreo de CO₂ en un aula de la Universidad Politécnica de Cataluña, ubicada en Terrasa Barcelona, las dimensiones de dicho recinto se presentan en la figura 3.37.

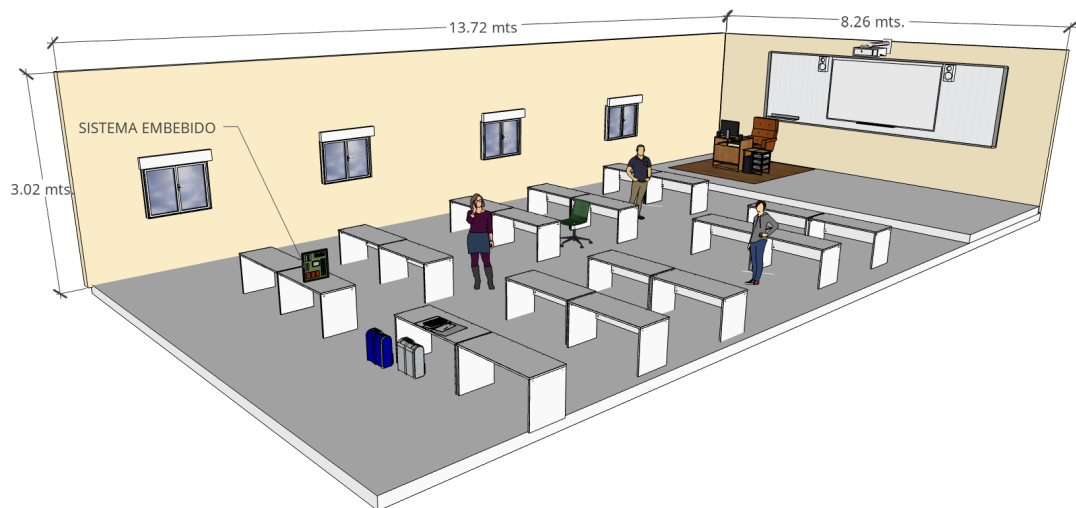


Figura 3.36: Salón de pruebas para el monitoreo del CO₂.

Se pone en marcha el sistema embebido a una temperatura de 22° centígrados a las 12:00 hora local del día 14 de marzo del 2023, la duración del evento es durante dos horas y se presentan la visualización de los datos en la *dashboard* ofrecidas por la *Thingspeak* en la figura 3.37.

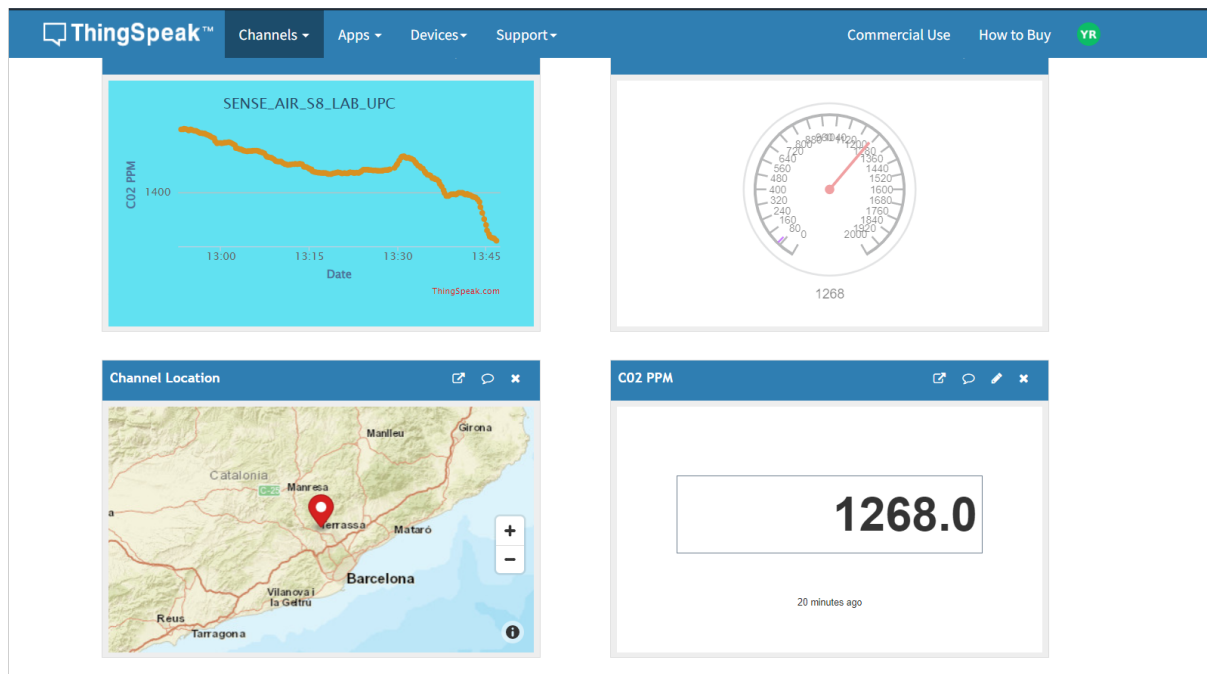


Figura 3.37: Salón de pruebas para el monitoreo del CO₂

Se realiza la recepción de datos durante 2 hrs. que tiene duración

una clase, el resultado mostrado en *Thinspeak* son visualizados con los siguientes eventos mostrados en la figura 3.38, y se describen a continuación

- Al inicio de la clase siendo 40 personas en el recinto con un horario de 12:16 hrs. se tiene un valor pico de 1923 ppm de CO₂ antes del nivel **A** se muestra un incremento de los valores por la reducción de los ocupantes.
- **A-C**: Se mantienen los niveles de CO₂ en promedio con un valor de 1625 ppm al mantener 31 ocupantes el aula, que es el periodo donde se lleva a cabo la clase en un horario de 12:25 a 13:35 hrs.
- **C-D**: Finaliza la clase a las 13:35 hrs donde la reducción de ocupantes a 15 alumnos provoca la disminución de las partículas de CO₂ a un promedio de 1394 ppm.
- **D-E**: El recinto concentra a 3 personas, a pesar de esto, los niveles de CO₂ se reducen lentamente, estando en un nivel crítico para la convivencia hasta las 13:54 hrs, indicando un periodo de 19 minutos para la reducción de CO₂.
- **E**: Los niveles de CO₂ son < 1000 ppm, los cuales son óptimos para la convivencia.

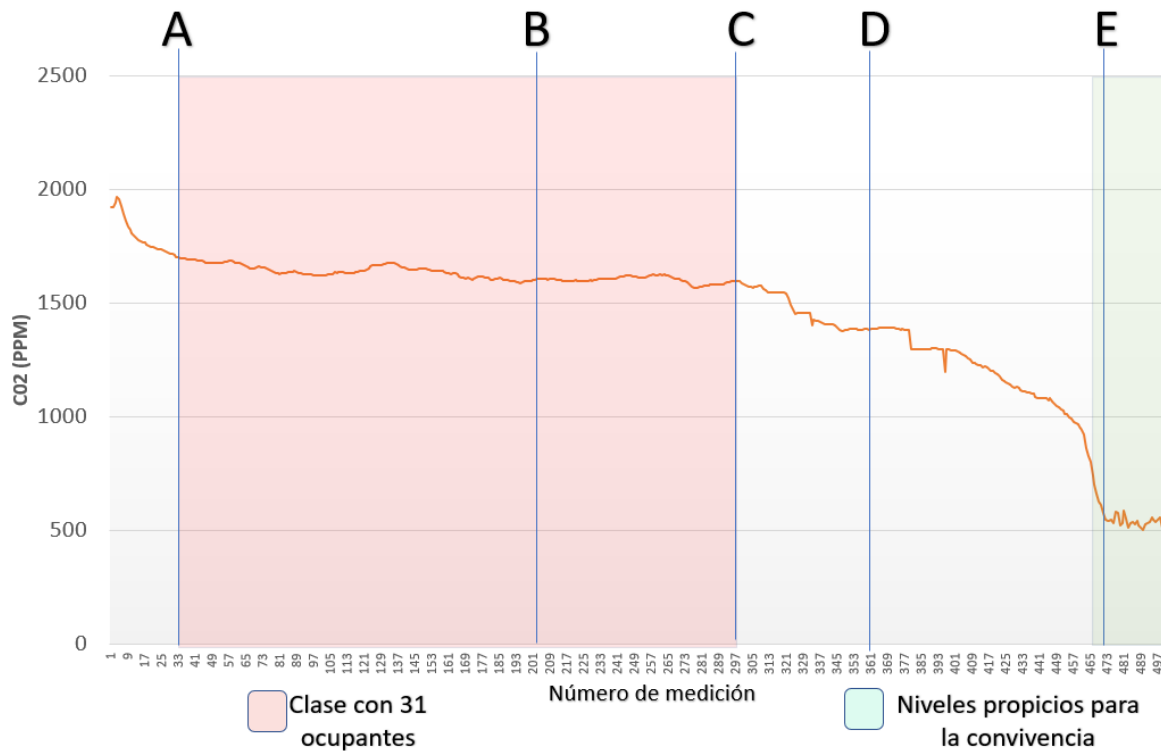


Figura 3.38: Salón de pruebas para el monitoreo del CO₂.

Se observa que la reducción de los niveles de CO₂ es gradual, y depende del área del recinto, por lo que la activación de dispositivos que puedan ayudar a mitigar niveles no propicios para la convivencia, ayudaría a acelerar el proceso de purificación del aire, lo cual se demuestra en un artículo escrito anteriormente por otras pruebas realizadas con el sistema embebido, en el artículo denominado *CO₂ monitoring system to warn of possible risk of spread of covid 19 in classrooms*, presentado en *el 2nd International Conference on Trends in Electronics and Health Informatics*, el cual será publicado en el mes de Diciembre del 2023 en actas de la conferencia por Springer Singapore Book Series: Lectures Notes in Networks and Systems (LNNS).

4. Análisis de datos

4.1. Estado del arte

Los estudios científicos indican que los seres humanos pueden tomar medidas efectivas para enfrentar las consecuencias de la contaminación atmosférica si cuentan con información previa sobre la evolución de las concentraciones de contaminantes. Por consiguiente, la predicción de estas concentraciones en la calidad del aire se ha convertido en un campo de investigación crucial para los investigadores del futuro. En comparación con la monitorización de los niveles de IAQ, la predicción es un enfoque rápido y económico para analizar las condiciones de salud ambiental.

Las condiciones de la calidad del aire tienen relación con cambios en la amplitud del movimiento respiratorio, aumento del flujo de sangre periférica y reducción del estado de funcionamiento de la corteza cerebral [67][68].

Son por estas razones que limitar los efectos de CO₂ son importantes, mantener los niveles óptimos que permanezcan por debajo de los umbrales dictados es posible, en ciertos casos con la puesta en marcha de equipos mecánicos, especialmente en ciudades metropolitanas donde el aire exterior contiene altas concentraciones de contaminantes atmosféricos[67][69].

La ventilación controlada por demanda DCV (*Distributed Control System*) basada en CO₂ ha atraído la atención inmediata de la industria de calefacción, ventilación y aire acondicionado (HVAC) desde su primera invención. Desde entonces, ha ido penetrando desde simples sistemas de una sola zona hasta configuraciones de sistemas complejos en edificios comerciales. Si bien se ha acumulado una gran cantidad de investigaciones sobre las aplicaciones de DCV, haciendo uso de métodos convencionales como el PID (Control el proporcional, integral, y derivativo) son usados, sin embargo es usual que estos controladores no capturen las no linealidades en las concentraciones de CO₂, volviéndose inestables[65][70].

Es importante mencionar que los datos de las variables medidas para el confort térmico en los edificios, cada vez son de fácil acceso, esto

debido a la conexión a dispositivos IoT y sensores altamente calibrados, esto ha dado pie al desarrollo de nuevas estrategias para el IAQ [71].

Un ejemplo de dichas técnicas es la mostrada en [71], la cual presenta una solución de gemelo digital para automatizar el monitoreo y el control de las emisiones de CO₂ de los activos existentes a través de la integración de IoT, modelos digitales e inteligencia artificial en una solución integral, una desventaja existente es debido al desarrollo de sistemas HVAC en edificios inteligentes es la complejidad del proceso de creación y mantenimiento de estos modelos. Los gemelos digitales requieren una gran cantidad de datos precisos y actualizados para reflejar con precisión el comportamiento del sistema HVAC en el edificio. Esto implica la necesidad de implementar y mantener una infraestructura de sensores y recopilar datos en tiempo real, lo que puede resultar costoso y complicado de gestionar.

Otra desventaja es la posibilidad de discrepancias entre el comportamiento del gemelo digital y el sistema HVAC real. Aunque los gemelos digitales se esfuerzan por reflejar fielmente el comportamiento del sistema, siempre existe la posibilidad de que existan factores no considerados o no capturados en el modelo digital. Esto puede conducir a decisiones incorrectas o subóptimas basadas en las predicciones del gemelo digital, lo que podría afectar negativamente la eficiencia y el rendimiento del sistema HVAC en el edificio.

Por otro lado el uso de IA, como lo son redes neuronales [72] o aprendizaje profundo [73][71] son técnicas cada vez más usadas, si bien las redes neuronales y el aprendizaje profundo son herramientas poderosas para la predicción en diversos campos, incluido el monitoreo del CO₂ en ambientes cerrados, también presentan algunas desventajas como lo son la necesidad de la obtención de grandes conjuntos de datos: Las redes neuronales y el aprendizaje profundo suelen requerir conjuntos de datos extensos para lograr un rendimiento óptimo.

Requieren poder computacional y recursos, el entrenamiento y la ejecución de redes neuronales y modelos de aprendizaje profundo pueden ser computacionalmente intensivos y requerir recursos significativos, como potencia de procesamiento y memoria. Esto puede ser una limitación en entornos con recursos limitados, especialmente si se necesita una respuesta en tiempo real o si se desea implementar en dispositivos con

capacidad computacional limitada.

El presente trabajo hace una diferenciación de otras investigaciones que tienen como objetivo maximizar la precisión de la predicción, generalmente haciendo uso de una base de datos recopilada durante varios días. El análisis de datos que se propone es realizar una correlación de diversas variables de confort ambiental para que de esta manera pronosticar el CO₂ en espacios interiores y un modelo de autoregresion lineal, ya que este gas representa uno de los contaminantes que más afecta la salud ambiental y la concentración de los individuos en espacios cerrados. Después de una recopilación rápida de datos, se realiza un entrenamiento previo del modelo propuesto sin necesidad de realizar algoritmos tan complejos los cuales requieren una alta capacidad de procesamiento de la información, lo anterior tiene como finalidad la predicción de niveles alto de CO₂ y así notificarle anticipadamente a los ocupantes que se encuentran en la habitación de un nivel de riesgo, esto también permitiría la activación de sistemas HVAC los cuales regulen los niveles de CO₂ en ambientes cerrados.

4.2. Power BI data y desarrollo de una *dashboard*

Después de recopilar los datos en la *dashboard* del *software IoT* denominado *Thingspeak*, enviados con formato .json hacia Microsoft Power BI para procesar y obtener una mejor visualización de los datos en el forma de gráficos, análisis de tendencias e inclusive pronósticos.

Power BI es una suite de herramientas de análisis y visualización de datos desarrollada por Microsoft. Permite a los usuarios conectar, transformar y visualizar datos de diversas fuentes en informes interactivos y paneles de control. Power BI se utiliza ampliamente en el análisis de datos y la generación de informes empresariales para obtener información y tomar decisiones informadas.

Power BI facilita la importación de datos desde una amplia gama de fuentes, como bases de datos, archivos de Excel, servicios en la nube, servicios web y muchas más. Una vez que los datos se han importado, se pueden transformar y modelar según las necesidades específicas utilizando la interfaz intuitiva de Power BI. Esto incluye la limpieza de datos, la combinación de varias fuentes de datos, la creación de relaciones entre

tablas y la definición de cálculos personalizados.

Después de modelar los datos, Power BI proporciona herramientas de visualización y análisis que permiten crear informes interactivos y paneles de control. Estos informes y paneles se pueden personalizar con una variedad de visualizaciones, como gráficos, tablas, mapas, indicadores clave de rendimiento (KPI) y muchas más. Los usuarios pueden explorar y filtrar los datos de manera interactiva, lo que les permite descubrir patrones, tendencias y relaciones ocultas en los datos.

Además de la creación de informes, Power BI también ofrece características de colaboración y compartición. Los informes y paneles se pueden publicar en línea y compartir con otros usuarios dentro o fuera de la organización. Esto facilita la colaboración en tiempo real y la toma de decisiones basada en datos.

Power BI es una herramienta poderosa para el análisis de datos que permite importar, transformar y visualizar datos de diversas fuentes. Ayuda a comprender mejor sus datos, identificar patrones y tendencias, y tomar decisiones informadas [74].

En el capítulo 3.13 se plantea visualización de los datos en la *dashboard* en *ThingSpeak*, se crea un proyecto el cual nos mostrara diversos datos de las variables sensadas del sistema embebido y apesar de que hacia uso de una *Dashboard* con las herramientas de *Thingspeak* el uso de *Power BI*, añade funcionalidades para el análisis de datos, a continuación se enumeran:

- Niveles máximos y mínimo de los últimos 100 datos.
- Mayor precisión en los datos de geolocalización del sistema.
- Promedio de los datos obtenidos.
- Obtención de los datos en tablas organizadas por herramientas de Excel.
- Predicción de los últimos 10 datos con obtención de confianza de 95 %.
- Visualización de la tendencia de los datos para los próximos 10 valores con fecha y hora.

Se presenta en la figura 4.1 la pagina 1 de la *dashboard* creada en *Power BI*, se puede visualizar los 4 campos registrados: Medición de

CO₂ en ppm, humedad relativa en porcentaje, Temperatura en grados centígrados y la presión atmosférica en hectopascales.

Se observa las últimas 100 mediciones con el registro de fecha y hora, así como los niveles máximos obtenidos.

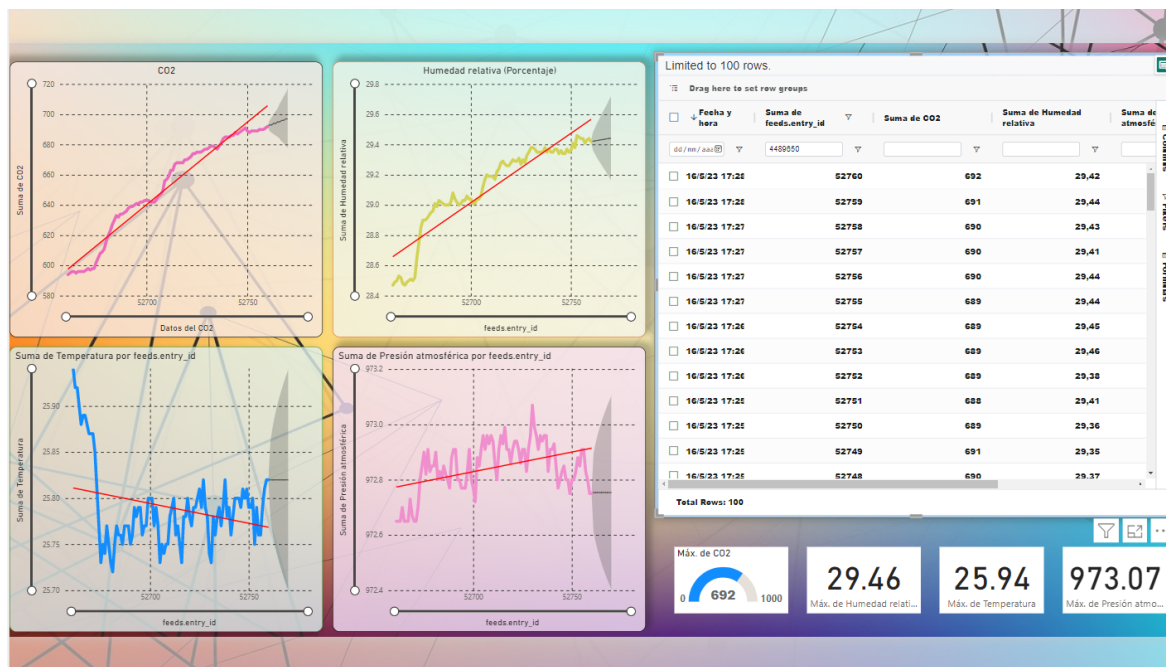


Figura 4.1: Página 1 de la *dashboard* generada en *Power BI*.

En la figura 4.2 se observa la pagina 2 de la *dashboard* creada en *Power BI*, se visualiza la geolocalización de los datos obtenidos, gráfica de CO₂ con la tendencia de los valores y una predicción de los 10 valores próximos.

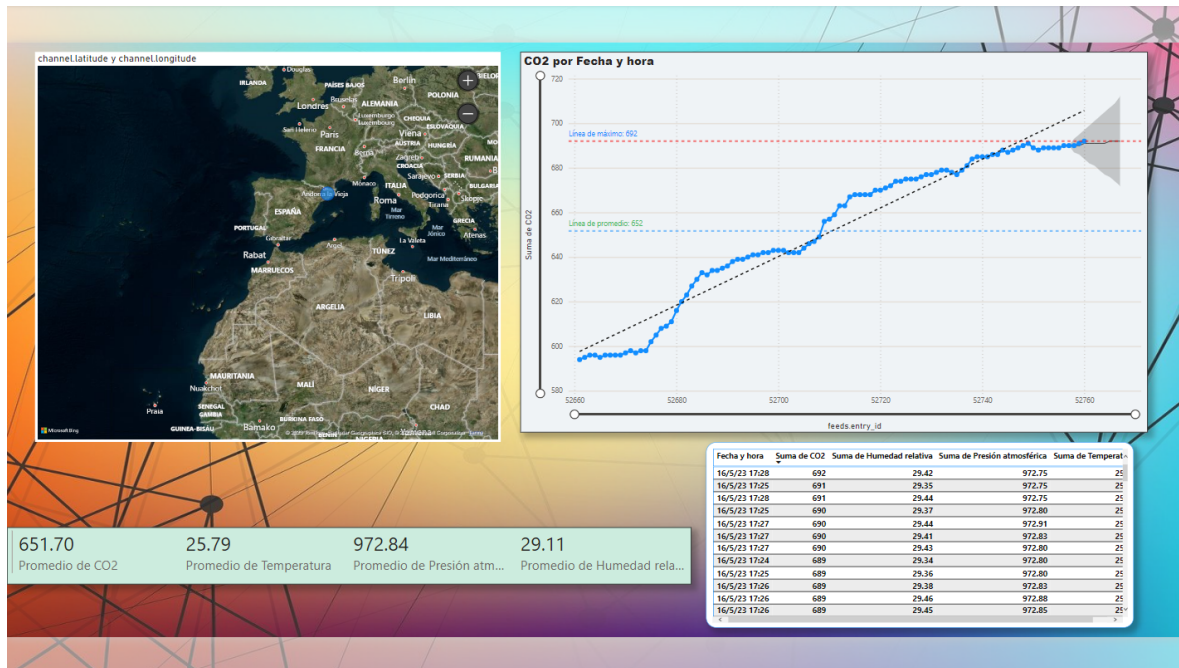


Figura 4.2: Página 2 de la *Dashboard* generada en *Power BI*.

4.3. Adición de sensado de nuevas variables

El sensor muestra la característica de realizar mediciones de tres parámetros por lo cual es una opción recomendable. Al ser un sistema embebido escalable y versátil la adición del sensor se coloca en el costado del la carcasa del sistema embebido, el cual se muestra en la figura 4.3.

El sensor que se adiciona en las actividades de la estancia MCIA, es el sensor BME280, el cual es un sensor de temperatura, la humedad relativa del aire y presión atmosférica, fabricado por la compañía Bosch Sensortec. El sensor utiliza una tecnología de compensación de temperatura y presión para mejorar la precisión de las mediciones, lo que significa que es capaz de proporcionar resultados precisos incluso en condiciones climáticas cambiantes.



Figura 4.3: Colocación del sensor añadido.

Es importante puntualizar que el sensor se coloca en una PCB exterior al sistema embebido, esto con la finalidad de darle mayor fiabilidad a las mediciones realizadas y sea mínima la afectación de temperatura generada por el mismo sistema embebido.

Se realiza el algoritmo para la lectura de las variables del sensor a través del sistema embebido y posteriormente plasmarlas en la pantalla la cuál cuenta el sistema embebido, esto es útil en caso de que no exista conexión a una red *WI-FI*, se actualiza el algoritmo para mostrar dos tipos de imágenes en la pantalla, la cual se presenta en la figura 4.4.

1. Muestra la medición de CO_2 y el estado actual de conexión a internet del sistema.
2. Muestra la medición de temperatura en grados centígrados, la humedad en porcentaje y la presión atmosférica en hectopascales.



Figura 4.4: Visualización de las imágenes mostradas en la pantalla del sistema embebido.

4.4. Desarrollo de algoritmos de análisis de datos

En comparación con el monitoreo de los niveles de calidad del aire interior (IAQ), la predicción es una forma eficiente y rentable de analizar las condiciones ambientales de salud. Para lograrlo, podemos utilizar sistemas de monitoreo de CO₂ con conectividad IoT para detectar las condiciones de riesgo de contagio de COVID-19 en tiempo real y, posteriormente, aplicar modelos estadísticos para prever las concentraciones de CO₂ en interiores. La idea principal consiste en establecer una relación entre las concentraciones medidas previamente y las concentraciones de contaminantes en eventos futuros. A continuación se presentan los algoritmos realizados y su relevancia con este proyecto de tesis.

4.5. Predicciones automatizadas con *Power BI*

Este método, basado en el algoritmo de suavización exponencial, permite realizar predicciones en series de tiempo. Se encuentra disponible en el panel de análisis del gráfico de línea, siempre y cuando la variable principal sea temporal. Mediante esta función, es posible identificar patrones en las series y replicarlos en el futuro, estableciendo un intervalo de confianza y señalando los puntos futuros que se desean predecir.

La aplicación de estrategias de análisis predictivo resulta fundamental para anticipar riesgos futuros y detectar posibles oportunidades de

mejora.

Si se cuenta con una estructura de datos sólida y las herramientas de análisis adecuadas, se podran realizar pronósticos probables en diversas áreas.

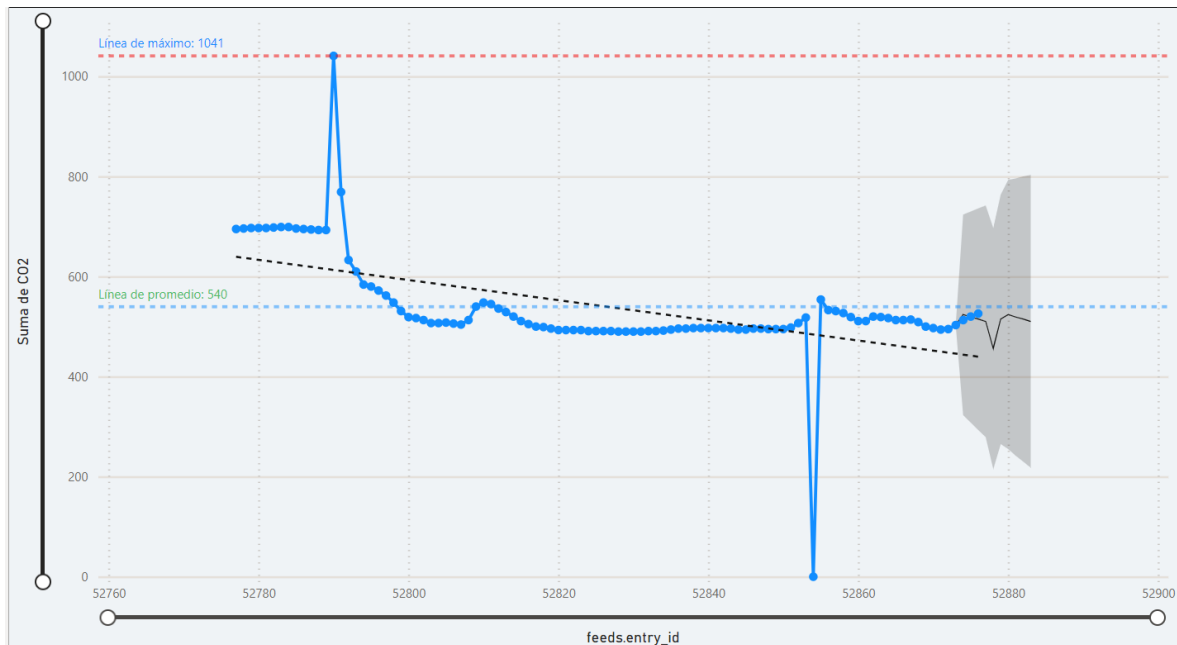


Figura 4.5: Predicción de valores de CO_2 en *Power BI*.

4.6. Modelo ARIMA

El modelo estadístico ARIMA es ampliamente utilizado para modelar y predecir series de tiempo [75][76], ya que considera la autoregresión, la integración y la media móvil. Este algoritmo resulta útil en ambientes cerrados donde las mediciones son temporalmente estacionarias, ya que permite modelar y predecir futuras concentraciones de CO_2 , lo que puede ser de gran importancia para tomar medidas preventivas y controlar la calidad del aire. Además, el uso de un modelo ARIMA permite analizar y predecir patrones de datos a largo plazo que podrían no ser evidentes a simple vista. De hecho, diversos estudios han demostrado que el algoritmo ARIMA resulta óptimo en la predicción de valores de CO_2 a corto plazo [75][77].

Cabe destacar que, debido a la sensibilidad de la variación de CO_2 a las condiciones ambientales momentáneas, en este estudio se limitó la ex-

ploración a un modelo que se puede entrenar con los valores registrados durante un intervalo pasado reciente (hasta 60 minutos antes del momento actual). Para ello, se ha implementado un algoritmo en Python, el cual se muestra el algoritmo en la figura 4.6.

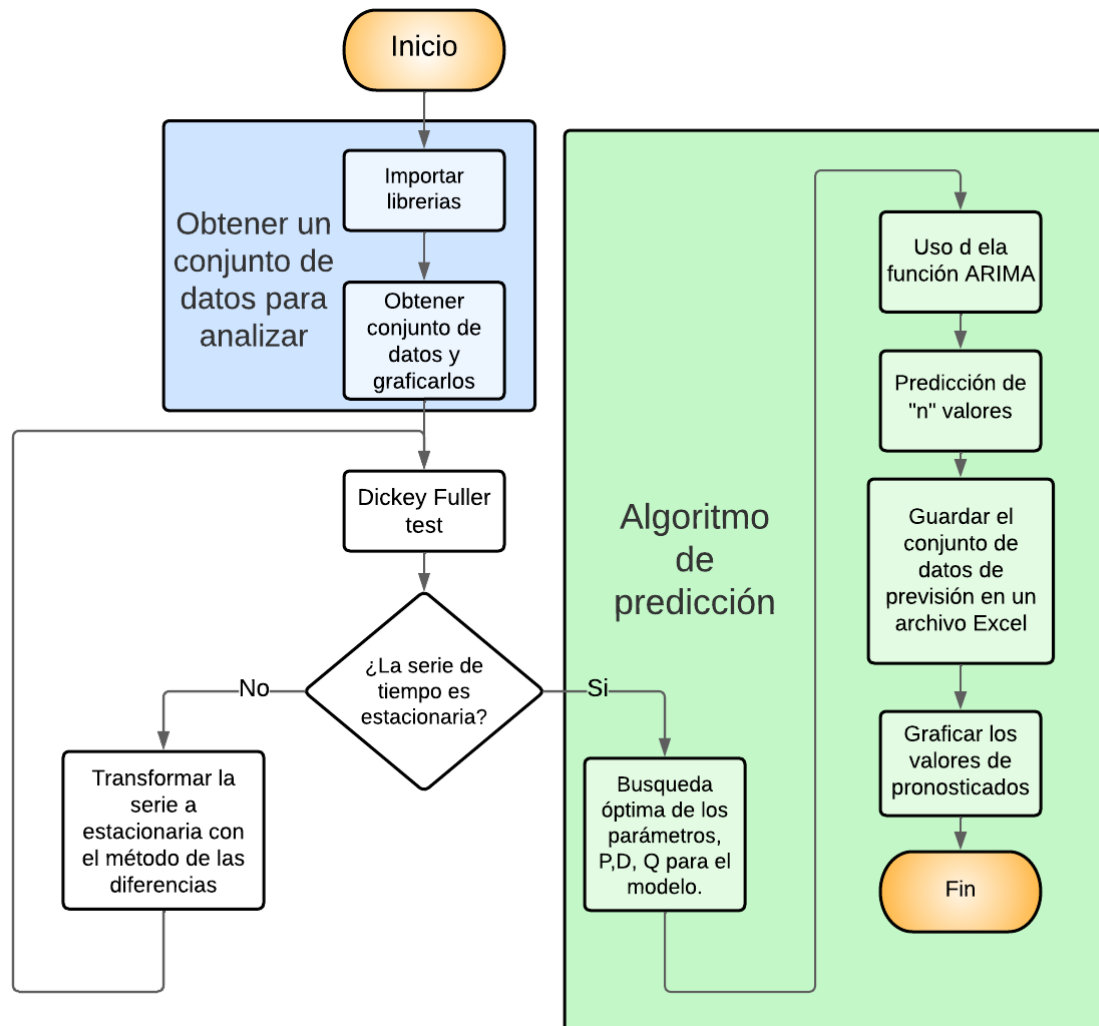


Figura 4.6: Algoritmo ARIMA implementado para la predicción de valores de CO₂.

Resultados del modelo ARIMA

El algoritmo implementado permite que el usuario ingrese el número de predicciones requeridas. Para ello, se utiliza la función de ARIMA en Python y se realiza un ciclo de predicciones que se almacena en un archivo con extensión .xlsx. En este estudio, se realizaron diez predicciones a partir del último valor observado, que corresponde al valor 2029. Por

tanto, las predicciones abarcan desde el valor 2030 hasta el 2039, como se muestra en la tabla 4.12. Además, se generó una gráfica que compara los valores observados con los pronosticados, la cual se presenta en la figura 4.7.

Mediciones	Pronóstico	Lectura real	Error
2030	591.5305259	592	0.469474
2031	595.3692465	594	1.36925
2032	597.1317144	595	2.13171
2033	598.0025478	597	1.00255
2034	598.4929647	598	0.49296
2035	598.8208878	598	0.82089
2036	599.0779173	599	0.07792
2037	599.3042425	600	0.695758
2038	599.5173853	599	0.51739
2039	599.7245392	599	0.72454

Tabla 4.1: Algoritmo ARIMA implementado para la predicción de valores de CO₂.

Se observa que el error máximo es de 2.139 PPM y el valor más cercano tiene una diferencia de .077 PPM lo cual representa un rango con un margen de error menor a 2.5 PPM lo cual podría ser un indicador valioso para predecir futuros valores. Se realiza la gráfica de los valores de predicción y los reales en la figura 4.7.

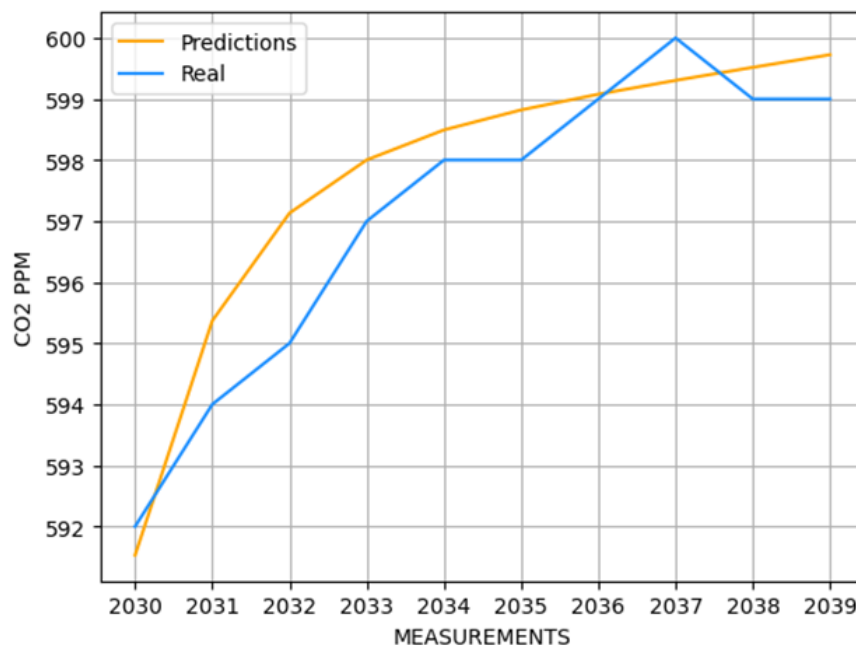


Figura 4.7: Comparación de valores reales y pronosticados de CO₂.

El modelo estadístico ARIMA ha demostrado ser una herramienta altamente útil para predecir los niveles de CO₂ en interiores. Además, los gráficos que comparan las predicciones con los valores reales muestran que el modelo tiene una buena capacidad para capturar tendencias y patrones estacionales en los próximos 15 datos. Sin embargo, es importante tener en cuenta que el modelo se basa en ciertos supuestos sobre la estacionariedad y la autocorrelación de los datos, lo que limita su aplicabilidad en situaciones donde estos supuestos no son válidos. En nuestro caso de estudio, se mostró una serie no temporal, por lo que se convirtió en una serie temporal utilizando técnicas de diferenciación. Si bien el modelo puede predecir valores próximos con precisión, se ha observado que su rendimiento disminuye significativamente cuando se le exige un número alto de predicciones. En general, el modelo ARIMA puede ser una herramienta valiosa para la predicción de los niveles de CO₂ y puede ayudar a tomar decisiones informadas para reducir los riesgos de contagio de COVID-19 en ambientes cerrados.

Cabe recalcar que este análisis fue presentado como artículo en la revista *International Journal of Combinatorial Optimization Problems and Informatics* edición 2023, el cual se presenta la portada del artículo en los apéndices 6.3.3.

4.7. Predicción de niveles de CO₂ en interiores

Tomando en cuenta el diagrama de bloques del sistema general mostrados en 1.1 se adiciona un nuevo bloque de análisis de datos, donde se obtendrá la predicción de los datos y la correlación de las variables de CO₂, temperatura, humedad relativa y presión atmosférica, por lo cual el diagrama de bloques que ejemplifica dicha adición se muestra en la figura 4.8

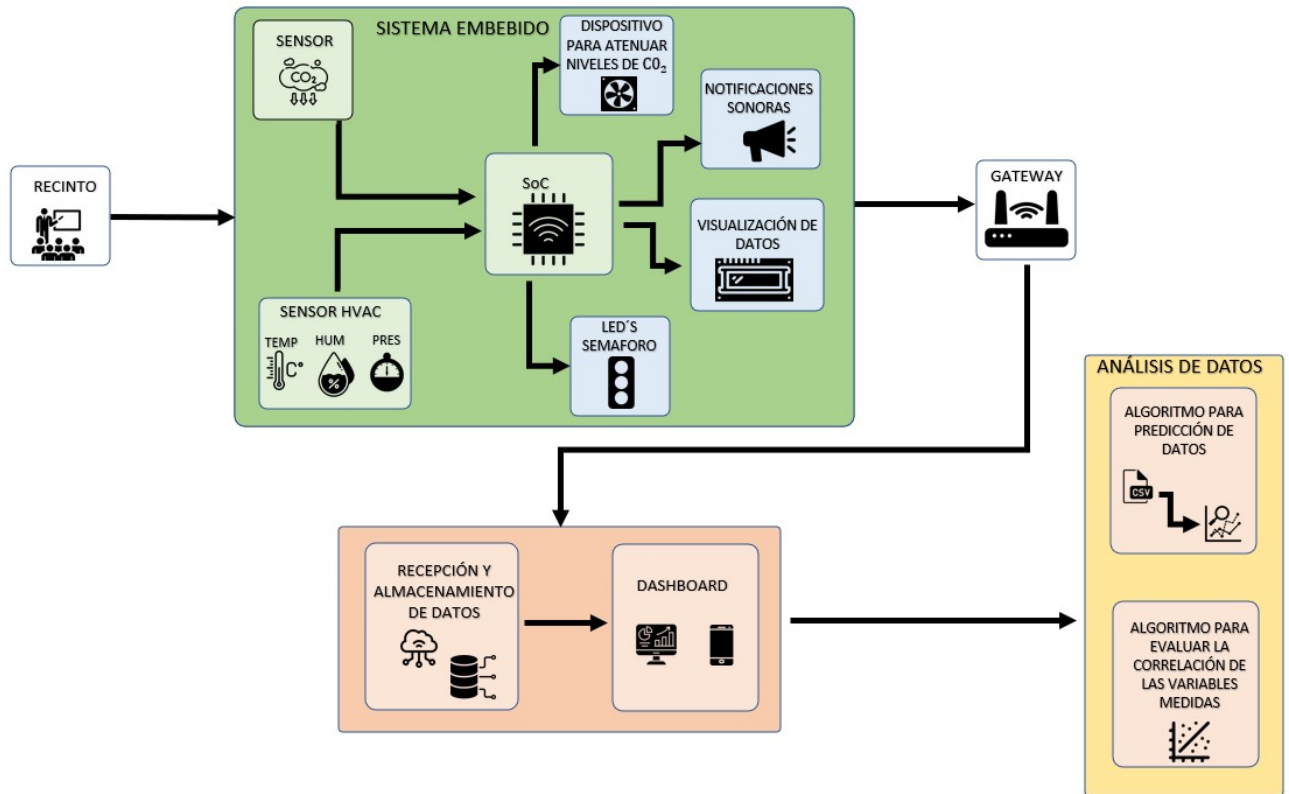


Figura 4.8: Diagrama de bloques con la adición del análisis de datos.

4.8. Correlación de las variables medidas: CO₂, temperatura, humedad relativa y presión atmosférica

Existen múltiples factores ambientales que pueden influir en los niveles de CO₂ en ambientes interiores cerrados, tales como la temperatura, la humedad y la presión atmosférica. La relación entre estos factores ha sido ampliamente estudiada en la literatura científica, y se ha encontrado que existe una correlación positiva entre los niveles de CO₂ y la temperatura y la humedad relativa del aire. Además, se ha demostrado que la presión atmosférica también puede influir en los niveles de CO₂, ya que una disminución en la presión atmosférica puede dar lugar a una mayor concentración de CO₂ en el aire interior. Por lo tanto, es importante considerar estos factores ambientales al monitorear los niveles de CO₂ en ambientes interiores cerrados y tomar medidas preventivas adecuadas para controlar su concentración y garantizar una buena calidad del aire interior, para evitar el riesgo de contagio de COVID-19 o incluso alguna enfermedad respiratoria cuya transmisión sea el aire o una mala calidad

del mismo [70].

En el caso de pronosticar la concentración de CO_2 en ambientes interiores cerrados, la regresión lineal puede ser una herramienta útil para modelar la relación entre la concentración de CO_2 y otras variables ambientales, como la temperatura, la humedad y la presión atmosférica.

La regresión lineal permite cuantificar la relación entre estas variables y la concentración de CO_2 , lo que puede ayudar a comprender mejor los factores que influyen en la calidad del aire interior. Además, una vez que se ha establecido una relación significativa, se pueden utilizar los valores de las variables ambientales para predecir la concentración futura de CO_2 .

Se hace uso de funciones de *Python*, Este código comienza importando varias librerías de Python que se utilizan para realizar análisis de datos y entrenar modelos de regresión lineal. Luego, carga un archivo CSV en un *DataFrame* de pandas llamado "df1" y lo modifica eliminando una columna y cambiando los nombres de las columnas restantes. Después, se seleccionan las columnas de interés y se asignan a diferentes variables.

A continuación, se crean cuatro gráficas de dispersión utilizando las variables seleccionadas. Cada gráfica representa la relación entre una variable y el CO_2 . Se ajusta el espacio entre las gráficas y se muestran en una sola figura utilizando el método "plt.show()".

Posteriormente se muestra la matriz de correlación entre las variables utilizando la función "heatmap" de la librería *Seaborn*. A continuación, se preparan los datos para entrenar un modelo de regresión lineal utilizando las variables Temperatura, humedad y presión atmosférica para predecir la concentración de CO_2 .

Se separan los datos en conjuntos de entrenamiento y prueba. Se ajusta un modelo de regresión lineal a los datos de entrenamiento y se evalúa su desempeño utilizando la función "score()". Se hacen predicciones con el modelo entrenado utilizando los datos de prueba y se calcula el error cuadrático medio entre las predicciones y los valores reales.

Por último, se guarda un archivo Excel con los valores reales y predichos de CO_2 y se muestra una gráfica de dispersión de la concentración de CO_2 versus las variables de temperatura, humedad y presión.

En resumen, este código carga un archivo CSV, muestra algunas gráficas de dispersión, calcula la matriz de correlación, entrena y evalúa un modelo de regresión lineal, realiza predicciones con el modelo entrenado y guarda un archivo Excel con los resultados, lo cual se muestra en la figura 4.9.

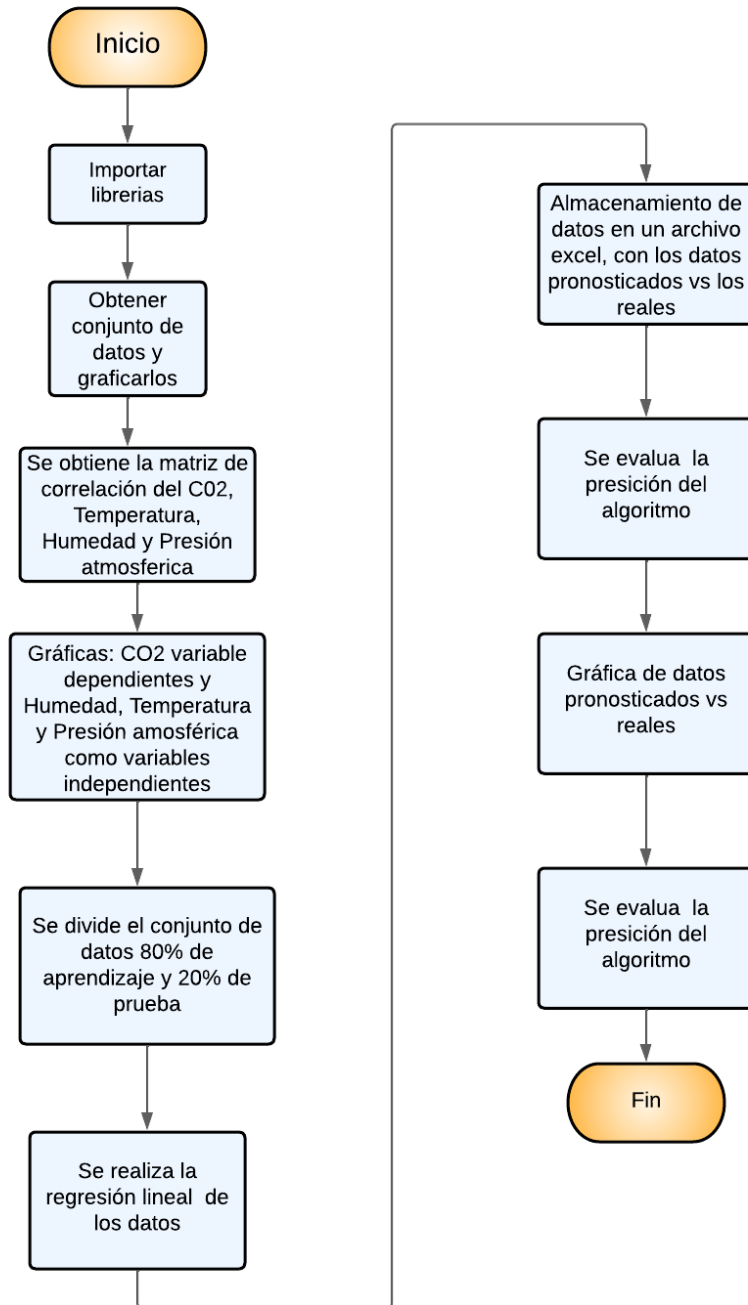


Figura 4.9: Algoritmo de regresión lineal implementado para la predicción de valores de CO₂.

Se realiza la gráfica de relación con la variable dependiente con CO₂ y las variables independientes como temperatura, humedad y presión atmosférica, se puede notar que existe una relación lineal entre estas variables, en la figura 4.10

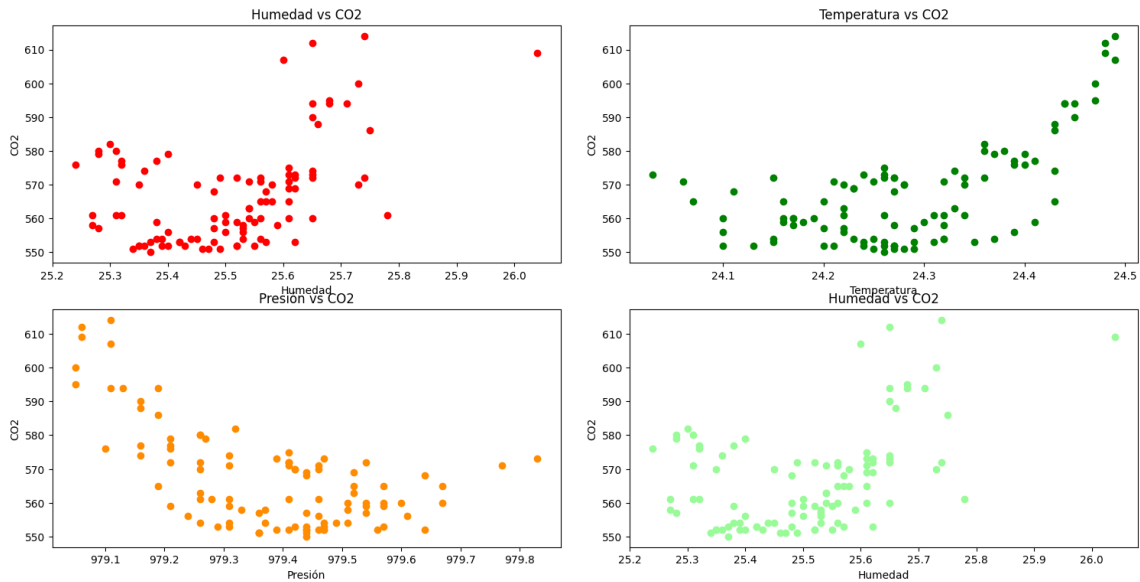


Figura 4.10: Matriz de correlación entre variables.

Se presenta el resultado del análisis realizado a partir de una matriz de correlación, esto es útil porque proporciona información valiosa sobre la relación entre varias variables. La matriz de correlaciones es una tabla cuadrada donde cada celda representa el coeficiente de correlación en nuestro caso de 4 variables.

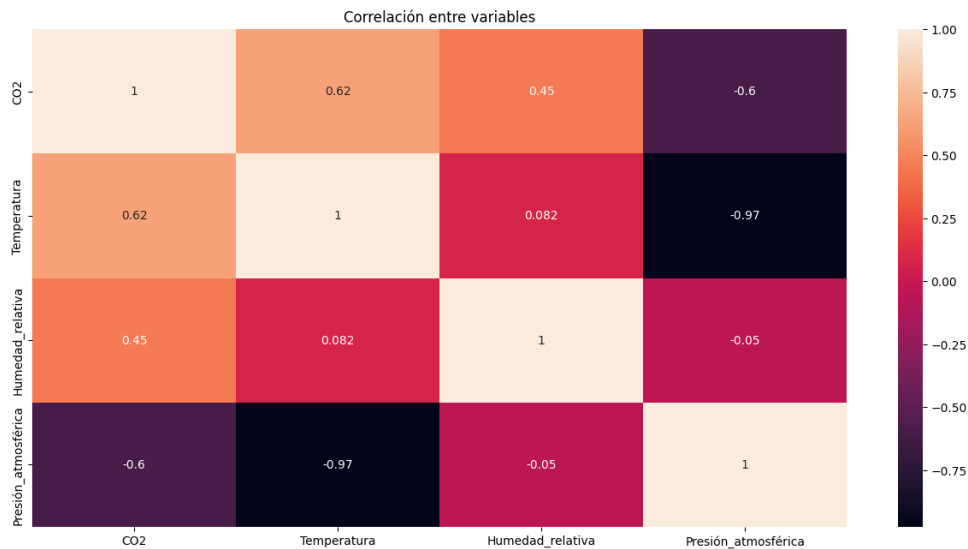


Figura 4.11: Matriz de correlación entre variables.

Los coeficientes de correlación pueden variar de -1 a 1, donde un valor de 1 indica una correlación positiva perfecta entre dos variables (es decir, que aumentan o disminuyen juntas en forma proporcional), un valor de -1 indica una correlación negativa perfecta entre dos variables (es decir, que una aumenta mientras la otra disminuye), y un valor de 0 indica que no hay correlación entre las dos variables.

Se presenta en la figura 4.11, la correlación de estos datos analizados de la base de datos recopilada del sensor dada por el sistema embebido.

Las correlaciones pueden ser visualizadas a través de índices o incluso representadas mediante colores intensos en caso de una correlación fuerte, y colores débiles en caso contrario.

Se observa una correlación moderada a fuerte, lo que sugiere que hay una relación significativa entre las variables de CO₂ y la temperatura. Una relación baja entre CO₂ y la Humedad relativa.

Con esto es posible realizar un pronóstico de los datos, en los cuales la variable dependiente es el CO₂ y la independiente una relación entre las tres variables de temperatura, humedad relativa y presión atmosférica, almacenándolos en un archivo CSV, el cual se muestra el resultado de la tabla generada en la 4.12

Co2 real	Co2 pronosticado	Error (PPM)
550	558	8
571	558	13
612	558	54
572	558	14
573	558	15
577	558	19
558	558	0
594	558	36
588	558	30
574	558	16
595	558	37
594	558	36
565	558	7
573	558	15
572	558	14
569	558	11
552	558	6
570	558	12
559	558	1
558	558	0
571	558	13
565	558	7
572	558	14
559	558	1
559	558	1

Figura 4.12: Comparación de los valores reales vs pronosticados.

Posteriormente se realiza un cálculo del error cuadrático medio entre los valores pronosticados y los reales, obteniendo un error de: **12.023**, mostrando un promedio de 12 ppm de error en un 80% de los datos de entrenamiento y un 20% de datos de prueba.

Este análisis fue presentado como cartel en el *XIII Congreso nacional de tecnología aplicada a ciencias de la salud & III Congreso internacional de tecnología aplicada a ciencias de la salud*, en Ciudad de México, México del 15 al 17 de Junio del 2023, el cual se presenta dicho cartel en los apéndices 6.3.3.

4.9. Conclusiones del análisis de datos

El modelo ARIMA el cual se plantea en el primer algoritmo, demostró ser una herramienta útil para la predicción del CO₂. Además, los gráficos de la predicción versus los valores reales indicaron una buena

capacidad del modelo para capturar las tendencias y patrones estacionales en los 15 datos próximos. Sin embargo, es importante tener en cuenta que el modelo se basa en supuestos sobre la estacionalidad y la autocorrelación de los datos, por lo que es importante tener cuidado al aplicarlo a situaciones en las que estos supuestos pueden no ser válidos. En nuestro caso de estudio se mostró una serie no temporal, por lo cual se convirtió la serie a temporal haciendo uso de diferencias. Si bien el algoritmo predice valores próximos, muestra un error considerable cuando se le exige un número alto de predicciones. En general, el modelo ARIMA puede ser una herramienta valiosa para la predicción de CO_2 y las posibles alternativas y decisiones que se pueden tomar en cuenta para la reducción de contagios de COVID-19.

El segundo algoritmo, muestra una correlación de variables monitoreadas para la calidad del aire en interiores, por lo cual obtener una estrecha relación de humedad, temperatura CO_2 y presión atmosférica, son una oportunidad para realizar un sistema de ventilación y aire acondicionado, y de esta forma, optimizar el consumo de energía. Por ejemplo, si el algoritmo detecta que los niveles de CO_2 son altos, puede indicar que se debe aumentar el flujo de aire fresco para mejorar la calidad del aire en el espacio cerrado.

Con los resultados obtenidos se deduce que cuando hay una acumulación de CO_2 en un espacio cerrado, puede haber un aumento en la temperatura debido a que el CO_2 actúa como un gas de efecto invernadero, atrapando el calor en el ambiente. Además, el aumento del CO_2 puede disminuir la cantidad de oxígeno disponible en el aire, lo que puede provocar una sensación de bochorno y afectar la calidad del aire.

Por otro lado, el aumento de la temperatura y la humedad pueden aumentar la producción de CO_2 , ya que esto puede acelerar la respiración y la transpiración de las personas y los seres vivos en el espacio cerrado. La presión atmosférica también puede afectar la cantidad de CO_2 presente en el aire, ya que los cambios en la presión pueden afectar la concentración de gases en el ambiente.

Es importante monitorear y mantener los niveles adecuados de CO_2 , temperatura, humedad y presión atmosférica en los espacios cerrados para garantizar un ambiente saludable y seguro para las personas que los ocupan, en nuestro caso evitar un riesgo de contagio de COVID-19,

sin embargo la extrapolación de este sistema puede ser conveniente para enfermedades respiratorias o asociadas a la mala calidad del aire, como lo son, neumonía, gripe estacionaria, enfermedades cardiovasculares, etc.

Las actividades realizadas en el instituto MCIA nos otorgaron un panorama sobre la formación científica en centros de investigación europeos donde la calidad del aire en interiores para el confort es un aspecto realmente importante para evitar afecciones. Por lo cual obtener información de las actividades que aquí se presentan, es una buena opción para extrapolarlo a nuestro país y realizar mejoras considerables en proyectos de este tipo.

Conclusiones

Los capítulos presentados anteriormente presentan un análisis de algunas de las aplicaciones *IoT* para el monitoreo de la calidad del aire, el creciente potencial, las oportunidades y los retos para el mejoramiento que presenta esta área de los sistemas embebidos *IoT*. La información mostrada proporciona un panorama sobre los desarrollos existentes en este campo y las mejoras que desarrollaremos a nuestro sistema en comparación de los ya existentes. Por otro lado, se muestra un análisis acerca de las limitaciones de los sistemas ya existentes, cómo lo son una calibración no adecuada, así como el procedimiento para realizarla y un tiempo reducido de ciclo de vida, varios sistemas de monitoreo basados en *IoT* no generan notificaciones sonoras o alertamientos a los usuarios para advertir sobre las altas concentraciones de contaminantes en ambientes cerrados.

Se han identificado las características esenciales de un sistema embebido destinado a operar como un sensor *IAQ* y un dispositivo *IoT*, capaz de transmitir información de manera precisa, respetando los límites de transmisión preestablecidos.

Se han formulado las directrices de diseño imprescindibles para lograr un sistema embebido altamente eficiente, priorizando la conformidad con los requisitos específicos del fabricante, a fin de garantizar un uso adecuado de los componentes empleados en la configuración electrónica.

Un análisis exhaustivo de los niveles de CO_2 óptimos para asegurar la habitabilidad de espacios cerrados se ha realizado, respaldado por reconocidas entidades de salud pública, tales como la OMS, OPS y ASHRAE.

La transmisión inalámbrica de datos a través de la nube ha sido sometida a una serie de rigurosas pruebas documentadas en este trabajo de tesis. Dichas pruebas han demostrado que el protocolo MQTT ofrece una sólida seguridad en la encriptación de mensajes y en la autenticación de los dispositivos.

La integración exitosa de los elementos anteriores culmina en un dispositivo que supera los desafíos planteados al inicio de este proyec-

to de tesis. Además, se propone la incorporación de diversos sensores adicionales para capturar variables que inciden en los niveles de CO₂ en entornos interiores. Estos datos, una vez recopilados, se someterán a análisis con el objetivo de anticipar valores que puedan afectar adversamente la calidad del ambiente, y su relación en aspectos importantes cómo lo son el clima, la hora, presión atmosférica y humedad relativa.

5. Bibliografía

- [1] Wei-jie Guan, Zheng-yi Ni, Yu Hu, Wen-hua Liang, Chun-quan Ou, Jian-xing He, Lei Liu, Hong Shan, Chun-liang Lei, David SC Hui, et al. Clinical characteristics of coronavirus disease 2019 in china. *New England journal of medicine*, 382(18):1708–1720, 2020.
- [2] A. Serrano-Cumplido, P.B. Antón-Eguía Ortega, A. Ruiz García, V. Olmo Quintana, A. Segura Fragoso, A. Barquilla Garcia, and Á. Morán Bayón. Covid-19. la historia se repite y seguimos tropezando con la misma piedra. *Medicina de Familia. SEMERGEN*, 46:48–54, 2020. COVID19 en Atención Primaria.
- [3] A de Velasco, Adriana Vadillo-Santos, P de Guevara, and Oscar Quiroz-Castro. Covid-19: historia actual de una pandemia y sus hallazgos clínico-radiológicos. In *Anales de Radiología México*, volume 19, pages 1–11, 2020.
- [4] Domingo Pérez Bejarano. Modos de transmisión y diseminación interhumana del virus sars-cov-2. *Revista de salud publica del Paraguay*, 11(1):87–101, 2021.
- [5] Ahmad Kamruzzaman Majumder, Abdullah Al Nayeem, Mahmu-da Islam, William S. Carter, Razib, and S. M. Munjurul Hannan Khan. Effect of covid-19 lockdown on air quality: Evidence from south asian megacities. *Environment & Natural Resources Journal*, 19(3):195 – 206, 2021.
- [6] Rajesh K. Bhagat, M. S. Davies Wykes, Stuart B. Dalziel, and P. F. Linden. Effects of ventilation on the indoor spread of covid-19. *Journal of Fluid Mechanics*, 903:F1, 2020.
- [7] National Collaborating Centre for Environmental Health. <https://www.ncceh.ca>, 13 de Enero de 2021.
- [8] Joanne Ktaz James M tielsh. Exposure to indoor biomass fuel and tobacco smoke and risk of adverse reproductive outcomes, mortality, respiratory morbidity and growth among newborn infants in south india. *International Journal of Epidemiology*, 38(5):1351–1363, 2009.

- [9] Particulate Matter (PM) Pollution. <https://www.epa.gov/pm-pollution/particulate-matter-pm-basics>, 2016.
- [10] Rajib Saha, S N M Azizul Hoque, M M R Manu, and Aminul Hoque. Monitoring air quality of dhaka using iot: Effects of covid-19. *2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*, pages 715–721, 2021.
- [11] Timotei Bidilă, Radu Nicolae Pietraru, Anca Daniela Ioniță, and Adriana Olteanu. Monitor indoor air quality to assess the risk of covid-19 transmission. *2021 23rd International Conference on Control Systems and Computer Science (CSCS)*, pages 356–361, 2021.
- [12] Daniel Natal Rodríguez. Espacios de trabajo inteligentes en la lucha contra la covid-19. *Archivo Digital UPM (Universidad Politécnica de Madrid)*, Enero-2021.
- [13] Internet of things for indoor air quality monitoring.
- [14] Organización Panamericana de la salud . <https://www.paho.org/es/temas/calidad-aire>, 10 de Mayo de 2022.
- [15] Samia Sultana. A comparative analysis of air pollution detection technique using image processing, machine learning and deep learning approach. *Global Journal of Computer Science and Technology: A Hardware & Computation*, 19(1), 2019.
- [16] Fulin Wang, Binruo Zhu, Rui Li, Dianshan Han, Zeyun Sun, Saejin Moon, Ziyang Gong, and Wenhong Yu. Smart control of indoor thermal environment based on online learned thermal comfort model using infrared thermal imaging. pages 924–925, 2017.
- [17] Gonçalo Marques Jagriti Saini, Maitreyee Dutta. A comprehensive review on indoor air quality monitoring systems for enhanced public health. *Conference on Automation Science and Engineering (CASE) Sustainable Environment Research*, (6):924–925, 2020.
- [18] Gonçalo Marques Jagriti Saini, Maitreyee Dutta. Biomass fuel combustion and health. *National Library of Medicine, National Center for biotechnology information*, (1):11–26, 1985.
- [19] L. Güette and M. Arocha. Diseño de un sistema inmótico mediante el uso del sistema embebido intel galileo. *IV Simposio Científico y Tecnológico en Computación*, 2016.

- [20] Natali Fiorella Alcorta Santusteban Jorge Luis Alva Alarcón. *Sistemas embebidos: Guía metodológica para su desarrollo*. Fondo editorial de la universidad privada Antenor Orrego, Octubre 2020. Primera edición.
- [21] Karrar S. Faraj. Printed circuit board manufacturing. 2020.
- [22] Autodesk inc. "pcb design made easy for every engineer" [https://www.autodesk.com/products/eagle/overview?term=1-YEAR&&\\$tab=subscription](https://www.autodesk.com/products/eagle/overview?term=1-YEAR&&$tab=subscription), 2022.
- [23] P.P. Shrimandilkar. Indoor air quality monitoring for human health. (1):7, 2013.
- [24] Padam Simkhada Sean Semple. Copd and chronic bronchitis risk of indoor air pollution from solid fuel: a systematic review and meta-analysis. *Thorax . Thorax 65(3)*, pages 321–328, 2019.
- [25] Lisa Thompson Daniel P. Pope, Vinod Mishra. Risk of low birth weight and stillbirth associated with indoor air pollution from solid fuel use in developing countries. *Epidemiologic Reviews*, (32):70–81, Abril 2010.
- [26] Y Ramesh Bhat, N Manjunath, D Sanjay, and Y Dhanya. Association of indoor air pollution with acute lower respiratory tract infections in children under 5 years of age. *Taylor & Francis*, 3(3):132–135, 2013.
- [27] Organización Panamericana de la Salud. ¿qué nos dice el monitoreo de la calidad del aire en el contexto del covid-19? <https://www.paho.org/es/documentos/que-nos-dice-monitoreo-calidad-aire-contexto-covid-19>, 23 Abril 2020.
- [28] Gökhan M. Mutlu Robert B. Hamanaka. Particulate matter air pollution: Effects on the cardiovascular system. *16 November 2018*, 16 Noviembre 2018.
- [29] BJ Smith, M Nitschke, LS Pilotto, RE Ruffin, DL Pisaniello, and KJ Willson. Health effects of daily indoor nitrogen dioxide exposure in people with asthma. *European Respiratory Journal*, 16(5):879–885, 2000.

- [30] U. Satish et al. Is CO_2 an indoor pollutant? direct effects of low-to-moderate CO_2 concentrations on human decision-making performance. *1*, 120(12):7, 2012.
- [31] Jason Seitz & Chenan Tong . Lmp91051 nDIR CO_2 gas detection system <https://www.ti.com/lit/an/snua207/snua207.pdf>, Mayo 2013.
- [32] Winsen-sensors. Infrared CO_2 sensor module (model: Mh-z19b) [https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-manual\(ver1_7\).pdf](https://www.winsen-sensor.com/d/files/infrared-gas-sensor/mh-z19b-co2-manual(ver1_7).pdf), 2020.
- [33] Senseair-sensors. Senseair s8 residencial) <https://senseair.com/products/size-counts/s8-residential/>, 2022.
- [34] Senseair. Senseair s8 residencial <https://senseair.com/products/size-counts/s8-residential/>, 2022.
- [35] Dylan Wall, Paul McCullagh, Ian Cleland, and Raymond Bond. Development of an internet of things solution to monitor and analyse indoor air quality. *Internet of Things*, 14:100392, 2021.
- [36] Espressif Systems. Esp32 series https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf, 2022.
- [37] Arduino S.r.l. Arduino uno rev 3.0” <https://store-usa.arduino.cc/products/arduino-uno-rev3?selectedStore=us>, 2021.
- [38] Arduino S.r.l. Arduino nano” <https://store-usa.arduino.cc/products/arduino-nano>, 2021.
- [39] Arduino S.r.l. Arduino uno wifi rev2” <https://store-usa.arduino.cc/products/arduino-uno-wifi-rev2>, 2021.
- [40] Raspberry. Raspberry pi 4, <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>, 2022.
- [41] Espressif. Productos espressif <https://www.espressif.com/en/products/socs/esp32>, 2022.
- [42] DigiKey. Desarrollar rápidamente aplicaciones basadas en micro-controladores en tiempo real utilizando micropython <https://onx.la/106a9>, 2022.

- [43] Damien George. Micropython <https://micropython.org>, 2022.
- [44] Amazon Inc. "thonny, python ide for beginners¿qué es un ide?" <https://aws.amazon.com/es/what-is/ide/>, 2022.
- [45] Thonny. "thonny, python ide for beginners" <https://thonny.org>, 2022.
- [46] R. Albalak N. Bruce, R. Perez-Padilla. Indoor air pollution in developing countries: a major environmental and public health challenge. 15(2):16, 2020.
- [47] J. Zhang. X. Yang, L. Yang. A wifi-enabled indoor air quality monitoring and control system: the design and control experiments. *IEEE International Conference on Control & Automation (ICCA)*, 13:6, 2017.
- [48] S. Anushya N.V.R. Kumar, C. Bhuvana. Comparison of zigbee and bluetooth wireless technologies-survey. *International Conference on Information Communication and Embedded Systems (ICICES)*, 13:1–4, Febrero 2017.
- [49] N. Naik. Choice of effective messaging protocols for iot systems: Mqtt, coap, amqp and http. *IEEE International Systems Engineering Symposium (ISSE)*, page 7, 2017.
- [50] ThingSpeak. Thingspeak for iot projects, <https://thingspeak.com>, 2022.
- [51] Karen Fortuna, Kelly Aschbrenner, Matthew Lohman, Jessica Brooks, Mark Salzer, Robert Walker, Lisa George, and Stephen Bartels. Smartphone ownership, use, and willingness to use smartphones to provide peer-delivered services: Results from a national online survey. *Psychiatric Quarterly*, 89, 12 2018.
- [52] Alessandro Redondi Edoardo Longo y Pietro Manzoni. Mqtt st a spanning tree protocol for distributed mqtt brokers. *Universidad Politécnica de Valencia, España*, 2016.
- [53] Pagina oficial MQTT. "mqtt: The standard for iot messaging" <https://mqtt.org>, 2022.
- [54] The MathWorks Inc. "thingspeak for iot projects" <https://thingspeak.com>, 2022.

- [55] Espressif Systems. "esp32: Hardware design guidelines" https://www.espressif.com/sites/default/files/documentation/esp32_hardware_design_guidelines_en.pdf, 2021.
- [56] ShenZhen Hi-Link ElectronicCo.,Ltd. "5w. ultra small series powermodule" https://datasheet.lcsc.com/szlcsc/1912111437_HI-LINK-HLK-5M03_C209906.pdf, Septiembre 2019.
- [57] ASIC [Billiton Ltd.]. "tp4056 1a standalone linear li-lon battery charger with thermal regulation in sop-8" <https://dlmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf>, Septiembre 2021.
- [58] HMSEMI [Shenzhen Huazhimei Semiconductor Co., Ltd]. "dw01" <https://pdf1.alldatasheet.com/datasheet-pdf/view/1150541/HMSEMI/DW01A.html>, Septiembre 2021.
- [59] Fortune Semicon. "data sheet fs8205" https://datasheet.lcsc.com/lcsc/1810110910_Fortune-Semicon-FS8205_C32254.pdf, Febrero 2022.
- [60] SHOUDING [SHOUDING Shouding Semiconductor]. "sdb628" <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132508/SHOUDING/SDB628.html>, Febrero 2022.
- [61] Senseair . "modbus on senseair s8" <https://rmtplusstoragesenseair.blob.core.windows.net/docs/Dev/publicerat/TDE2067.pdf>, 2022.
- [62] Digi-key. "calculadora de ancho de trazas de pcb" <https://www.digikey.es/es/resources/conversion-calculators/conversion-calculator-pcb-trace-width>, 2023.
- [63] G. Marques J. Saini, M. Dutta. Espacios de trabajo inteligentes en la lucha contra la covid-19. *Archivo Digital UPM (Universidad Politécnica de Madrid)*, Enero-2021.
- [64] G. Marques J. Saini, M. Dutta. A comprehensive review on indoor air quality monitoring systems for enhanced public health. *BMC*, 30(1),6, 2020/01/29.

- [65] Servet Soyguder and Hasan Alli. An expert system for the humidity and temperature control in hvac systems using anfis and optimization with fuzzy modeling approach. *Energy and Buildings*, 41(8):814–822, 2009.
- [66] Adobe reader. "stl files." <https://www.adobe.com/creativecloud/file-types/image/vector/stl-file.html>, 2023.
- [67] Patrick Nzivugira Duhirwe, Jack Ngarambe, and Geun Young Yun. Energy-efficient virtual sensor-based deep reinforcement learning control of indoor co2 in a kindergarten. *Frontiers of Architectural Research*, 12(2):394–409, 2023.
- [68] Kenichi Azuma, Naoki Kagi, U. Yanagi, and Haruki Osawa. Effects of low-level inhalation exposure to carbon dioxide in indoor environments: A short review on human health and psychomotor performance. *Environment International*, 121:51–56, 2018.
- [69] Xing Lu, Zhihong Pang, Yangyang Fu, and Zheng O’Neill. The nexus of the indoor co2 concentration and ventilation demands underlying co2-based demand-controlled ventilation in commercial buildings: A critical review. *Building and Environment*, 218:109116, 2022.
- [70] A.I. Dounis and C. Caraiscos. Advanced control systems engineering for energy and comfort management in a building environment—a review. *Renewable and Sustainable Energy Reviews*, 13(6):1246–1261, 2009.
- [71] Arva Arsiwala, Faris Elghaish, and Mohammed Zoher. Digital twin with machine learning for predictive monitoring of co2 equivalent from existing buildings. *Energy and Buildings*, 284:112851, 2023.
- [72] J-P. Skön, M. Johansson, M. Raatikainen, K. Leiviskä, and M. Kolehmainen. Modelling Indoor Air Carbon Dioxide (CO2) Concentration using Neural Network, January 2022.
- [73] Jaehyun Ahn, Dongil Shin, Kyuho Kim, and Jihoon Yang. Indoor air quality analysis using deep learning with sensor data. *Sensors*, 17(11).

- [74] Microsoft company inc. Cree informes sofisticados y comparte conocimientos que impulsen los resultados — power bi. https://1c.cx/s_c0qG. (Accessed on 07/04/2023).
- [75] Denis LALANNE Hamed Alavi, Sailin Zhong. Predictive models of indoor carbon dioxide concentration to prevent daily decay of productivity and well-being in shared offices. *SmartPhil@ IUI*, pages 59–68, 2020.
- [76] Katherine A. Kiernan, Hao Wu, Ray Mugno, and Yulei Pang. Application of time series analysis methods to assess the impact of covid-19 on carbon dioxide emission reduction in connecticut. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 4513–4519, 2022.
- [77] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 1394–1401, 2018.

6. Apéndices

6.1. Esquemáticos diseñados

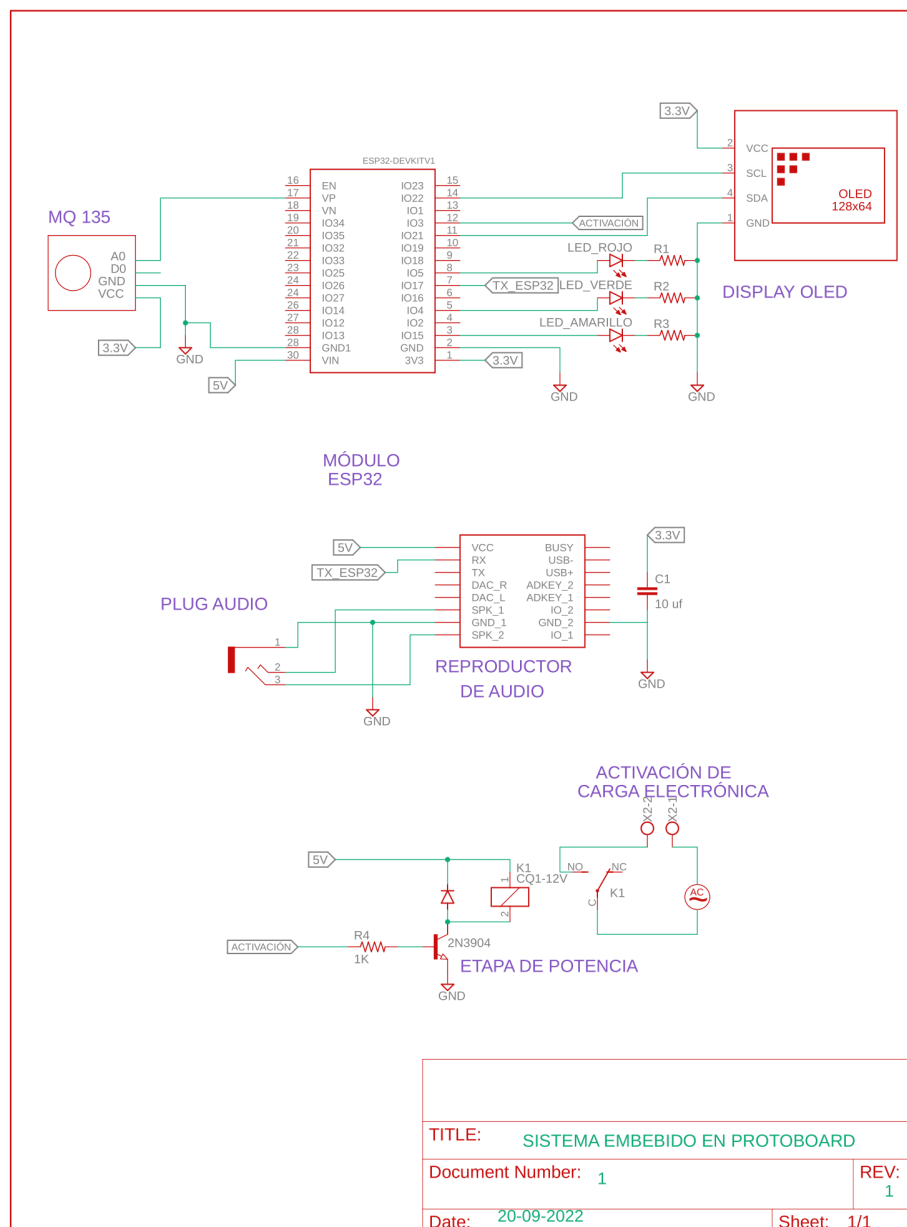


Figura 6.1: Esquemático del sistema prueba montado en protoboard.

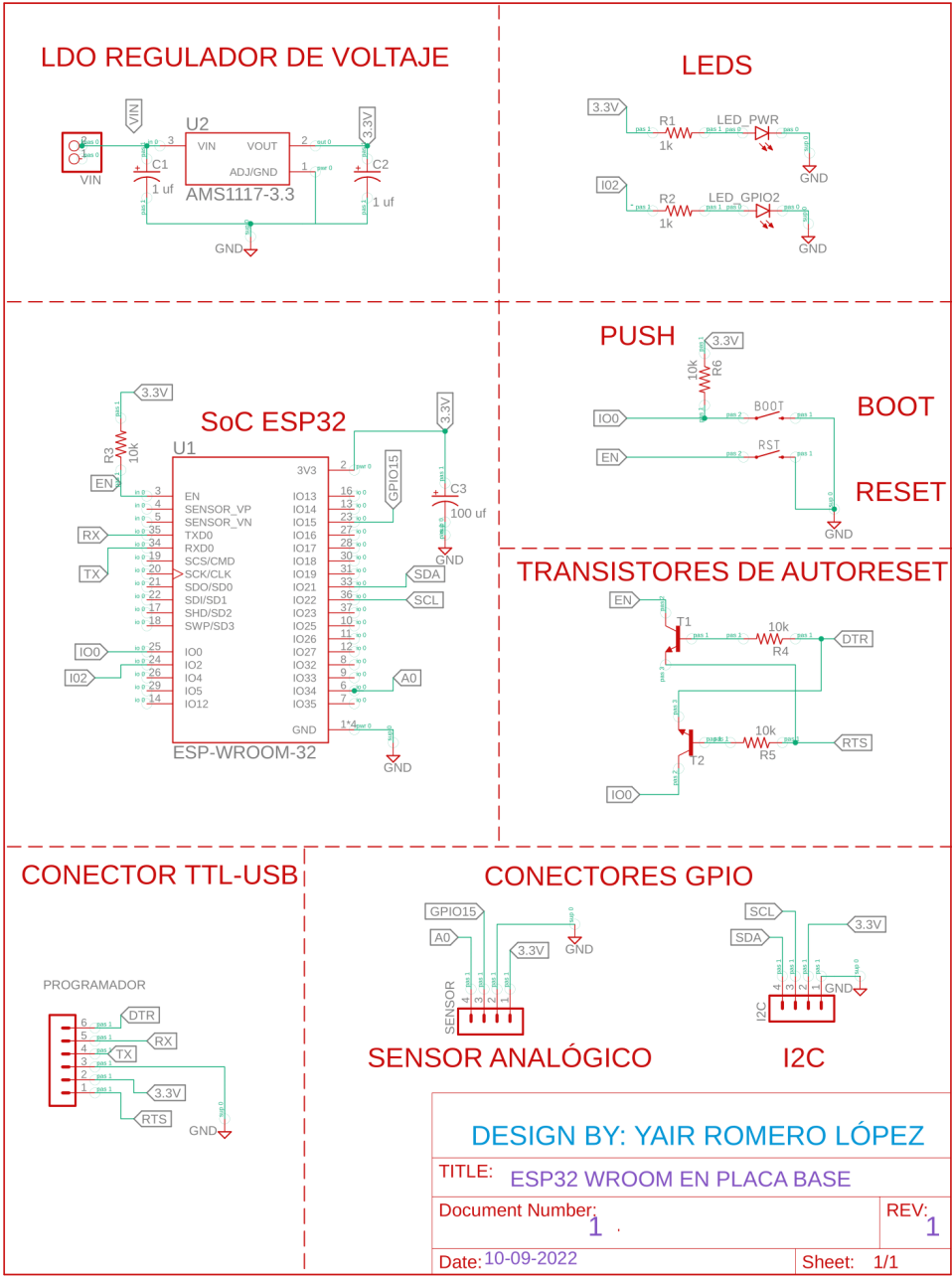


Figura 6.2: Esquemático ESP32 WROOM en placa base.

SENSOR DE CO2 CON CONEXIÓN A INTERNET

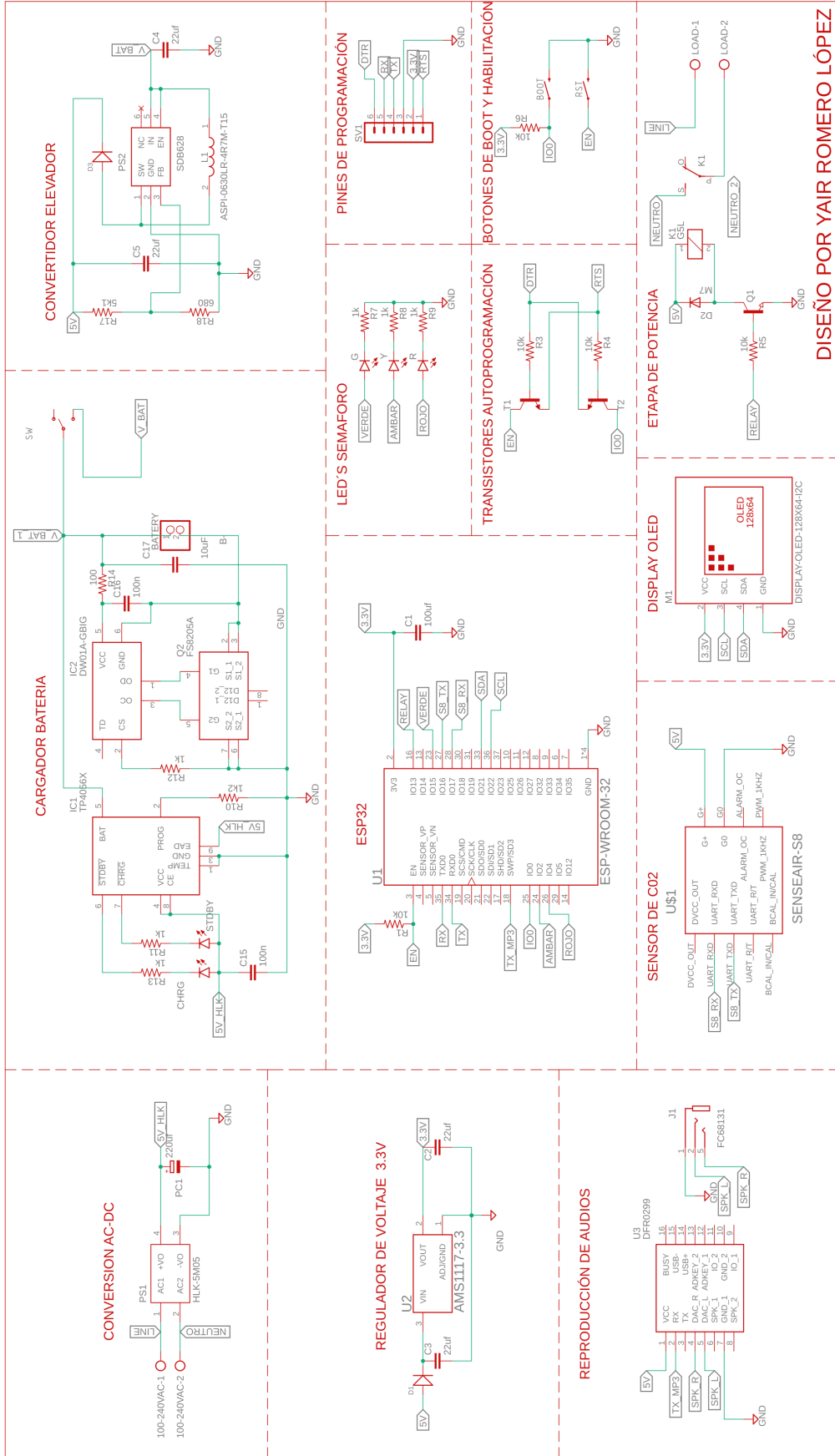


Figura 6.3: Esquemático final del sistema embebido.

6.2. Materiales usados para el diseño de la placa electrónica

CANTIDAD	DISPOSITIVO	EMPAQUETADO	DESCRIPCIÓN
3	AK500/2	AK500/2	CONECTOR TERMINAL BLOCK
3	STNDBY	LEDCHIPLED_0603	LED AZUL
3	CHRG	LEDCHIPLED_0603	LED NARANJA
3	G	LEDCHIPLED_0603	LED VERDE
3	Y	LEDCHIPLED_0603	LED AMARILLO
3	R	LEDCHIPLED_0603	LED ROJO
8	T1, T2, Q1	SOT23	NPN TRANSISTOR MMBT3904
2	Batery, SV1	PINHD-1X2, MA01	CONECTOR MACHO
4	SW	SWITCH-SPDT-SMD	
3	R14	R-US_R0603	Resistor 100 Ω
3	C15, C16	C-EUC0402	Capacitor 100nF
3	C1	C-EUC1206	Capacitor 100uF
15	R1, R3, R4, R5, R6	R-US_R0603	Resistor 10k Ω
3	C17	C-EUC0402	Capacitor 10uF
18	R7, R8, R9, R11, R12, R13	R-US_R0603	RESISTOR 1k Ω
3	R10	R-US_R0603	RESISTOR 1.2k Ω
3	PC1	CPOL-EUD	220uF
6	C2, C3	C-EUC0603	22uF
6	C4, C5	C-EUC1206	22uF
3	R17	R-US_R0603	RESISTOR 5.1 k Ω
3	R18	R-US_R0604	680 Ω
3	U2	AMS1117-3.3	SOT229P700X180-4N
3	L1	ASPI-0603	ASPI-0630LR 4.7 uH
3	U3	DIPS1778W50P254L2100H170Q16N	REPRODUCTOR DE AUDIO DFR0299
3	M1		DISPLAY-OLED-128X64-I2C
3	IC2	SOT95P280X145-6N	DW01A-GBIG
3	U1	MODULO_ESP-WROOM-32	MICROCONTROLADOR/WIFI/BLUETOOTH
3	J1		Conector audio FC68131
3	Q2	SOP65P640X120-8N	FS8205A
3	K1		RELEVADOR G5L
3	PS1		HLK-5M05
3	D2	DIOM505270X240	Diode SMD
6	BOOT,RESET	MOMENTARY-SWITCH-SPST-SMD-6.0X3.5MM	SWITCH
3	PS2	SDB628	SOT95P280X145-6N
2	U\$1	Senseair-s8	S8
3	D1, D3	DO-214AC	SMD-DIODO-SCHOTTKY-40V-1A(DO-214AC)
3	IC1	SOIC127P600X175-9N	TP4056 CARGADOR DE BATERÍA

6.3. Algoritmos realizados en *MicroPhyton*

6.3.1. Programa para la comunicación serial del Senseair S8 y el ESP-32

```
from machine import Pin, UART
uart = UART(1, tx=1, rx=3)
uart.init(9600, bits=8, parity=None, stop=1)
led = Pin(2, Pin.OUT)

while True:
    uart.write(b"\xFE\x44\x00\x08\x02\x9F\x25")
    time.sleep(1)
    resp= uart.read(7)
    high = ord (resp[3])
    low = ord (resp[4])
    c= (high*256)+low
    print(c)
```

6.3.2. Programa para el uso del sensor BME280

```
from machine import I2C
import time

# BME280 default address.
BME280_I2CADDR = 0x76

# Operating Modes
BME280_OSAMPLE_1 = 1
BME280_OSAMPLE_2 = 2
BME280_OSAMPLE_4 = 3
BME280_OSAMPLE_8 = 4
BME280_OSAMPLE_16 = 5

# BME280 Registers

BME280_REGISTER_DIG_T1 = 0x88 # Trimming parameter registers
```

```
BME280_REGISTER_DIG_T2 = 0x8A
BME280_REGISTER_DIG_T3 = 0x8C
```

```
BME280_REGISTER_DIG_P1 = 0x8E
BME280_REGISTER_DIG_P2 = 0x90
BME280_REGISTER_DIG_P3 = 0x92
BME280_REGISTER_DIG_P4 = 0x94
BME280_REGISTER_DIG_P5 = 0x96
BME280_REGISTER_DIG_P6 = 0x98
BME280_REGISTER_DIG_P7 = 0x9A
BME280_REGISTER_DIG_P8 = 0x9C
BME280_REGISTER_DIG_P9 = 0x9E
```

```
BME280_REGISTER_DIG_H1 = 0xA1
BME280_REGISTER_DIG_H2 = 0xE1
BME280_REGISTER_DIG_H3 = 0xE3
BME280_REGISTER_DIG_H4 = 0xE4
BME280_REGISTER_DIG_H5 = 0xE5
BME280_REGISTER_DIG_H6 = 0xE6
BME280_REGISTER_DIG_H7 = 0xE7
```

```
BME280_REGISTER_CHIPID = 0xD0
BME280_REGISTER_VERSION = 0xD1
BME280_REGISTER_SOFTRESET = 0xE0
```

```
BME280_REGISTER_CONTROL_HUM = 0xF2
BME280_REGISTER_CONTROL = 0xF4
BME280_REGISTER_CONFIG = 0xF5
BME280_REGISTER_PRESSURE_DATA = 0xF7
BME280_REGISTER_TEMP_DATA = 0xFA
BME280_REGISTER_HUMIDITY_DATA = 0xFD
```

```
class Device:
```

```
    """Class for communicating with an I2C device.
```

```
    Allows reading and writing 8-bit, 16-bit, and byte array values to
    registers on the device."""
```

```

def __init__(self, address, i2c):
    """Create an instance of the I2C device at the specified address using
    the specified I2C interface object."""
    self._address = address
    self._i2c = i2c

def writeRaw8(self, value):
    """Write an 8-bit value on the bus (without register)."""
    value = value & 0xFF
    self._i2c.writeto(self._address, value)

def write8(self, register, value):
    """Write an 8-bit value to the specified register."""
    b=bytearray(1)
    b[0]=value & 0xFF
    self._i2c.writeto_mem(self._address, register, b)

def write16(self, register, value):
    """Write a 16-bit value to the specified register."""
    value = value & 0xFFFF
    b=bytearray(2)
    b[0]= value & 0xFF
    b[1]= (value>>8) & 0xFF
    self.i2c.writeto_mem(self._address, register, value)

def readRaw8(self):
    """Read an 8-bit value on the bus (without register)."""
    return int.from_bytes(self._i2c.readfrom(self._address, 1),'little')
    & 0xFF

def readU8(self, register):
    """Read an unsigned byte from the specified register."""
    return int.from_bytes(
        self._i2c.readfrom_mem(self._address, register, 1),'little') & 0xFF

def readS8(self, register):
    """Read a signed byte from the specified register."""

```

```

    result = self.readU8(register)
    if result > 127:
        result -= 256
    return result

def readU16(self, register, little_endian=True):
    """Read an unsigned 16-bit value from the specified register, with the
    specified endianness (default little endian, or least significant byte
    first)."""
    result = int.from_bytes(
        self._i2c.readfrom_mem(self._address, register, 2), 'little')
        & 0xFFFF
    if not little_endian:
        result = ((result << 8) & 0xFF00) + (result >> 8)
    return result

def readS16(self, register, little_endian=True):
    """Read a signed 16-bit value from the specified register, with the
    specified endianness (default little endian, or least significant byte
    first)."""
    result = self.readU16(register, little_endian)
    if result > 32767:
        result -= 65536
    return result

def readU16LE(self, register):
    """Read an unsigned 16-bit value from the specified register, in little
    endian byte order."""
    return self.readU16(register, little_endian=True)

def readU16BE(self, register):
    """Read an unsigned 16-bit value from the specified register, in big
    endian byte order."""
    return self.readU16(register, little_endian=False)

def readS16LE(self, register):
    """Read a signed 16-bit value from the specified register, in little
    endian byte order."""

```

```

    return self.readS16(register, little_endian=True)

def readS16BE(self, register):
    """Read a signed 16-bit value from the specified register, in big
    endian byte order."""
    return self.readS16(register, little_endian=False)

class BME280:
    def __init__(self, mode=BME280_OSAMPLE_1, address=BME280_I2CADDR,
                 i2c=None, **kwargs):
        # Check that mode is valid.
        if mode not in [BME280_OSAMPLE_1, BME280_OSAMPLE_2, BME280_OSAMPLE_4,
                        BME280_OSAMPLE_8, BME280_OSAMPLE_16]:
            raise ValueError(
                'Unexpected mode value {0}. Set mode to one of '
                'BME280_ULTRALOWPOWER, BME280_STANDARD, BME280_HIGHRES, or '
                'BME280_ULTRAHIGHRES'.format(mode))
        self._mode = mode
        # Create I2C device.
        if i2c is None:
            raise ValueError('An I2C object is required.')
        self._device = Device(address, i2c)
        # Load calibration values.
        self._load_calibration()
        self._device.write8(BME280_REGISTER_CONTROL, 0x3F)
        self.t_fine = 0

    def _load_calibration(self):

        self.dig_T1 = self._device.readU16LE(BME280_REGISTER_DIG_T1)
        self.dig_T2 = self._device.readS16LE(BME280_REGISTER_DIG_T2)
        self.dig_T3 = self._device.readS16LE(BME280_REGISTER_DIG_T3)

        self.dig_P1 = self._device.readU16LE(BME280_REGISTER_DIG_P1)
        self.dig_P2 = self._device.readS16LE(BME280_REGISTER_DIG_P2)
        self.dig_P3 = self._device.readS16LE(BME280_REGISTER_DIG_P3)
        self.dig_P4 = self._device.readS16LE(BME280_REGISTER_DIG_P4)

```

```

self.dig_P5 = self._device.readS16LE(BME280_REGISTER_DIG_P5)
self.dig_P6 = self._device.readS16LE(BME280_REGISTER_DIG_P6)
self.dig_P7 = self._device.readS16LE(BME280_REGISTER_DIG_P7)
self.dig_P8 = self._device.readS16LE(BME280_REGISTER_DIG_P8)
self.dig_P9 = self._device.readS16LE(BME280_REGISTER_DIG_P9)

    self.dig_H1 = self._device.readU8(BME280_REGISTER_DIG_H1)
self.dig_H2 = self._device.readS16LE(BME280_REGISTER_DIG_H2)
    self.dig_H3 = self._device.readU8(BME280_REGISTER_DIG_H3)
    self.dig_H6 = self._device.readS8(BME280_REGISTER_DIG_H7)

h4 = self._device.readS8(BME280_REGISTER_DIG_H4)
h4 = (h4 << 24) >> 20
self.dig_H4 = h4 | (self._device.readU8(BME280_REGISTER_DIG_H5) &
0x0F)

h5 = self._device.readS8(BME280_REGISTER_DIG_H6)
h5 = (h5 << 24) >> 20
self.dig_H5 = h5 | (
    self._device.readU8(BME280_REGISTER_DIG_H5) >> 4 & 0x0F)

def read_raw_temp(self):
    """Reads the raw (uncompensated) temperature from the sensor."""
    meas = self._mode
    self._device.write8(BME280_REGISTER_CONTROL_HUM, meas)
    meas = self._mode << 5 | self._mode << 2 | 1
    self._device.write8(BME280_REGISTER_CONTROL, meas)
    sleep_time = 1250 + 2300 * (1 << self._mode)

    sleep_time = sleep_time + 2300 * (1 << self._mode) + 575
    sleep_time = sleep_time + 2300 * (1 << self._mode) + 575
    time.sleep_us(sleep_time) # Wait the required time
    msb = self._device.readU8(BME280_REGISTER_TEMP_DATA)
    lsb = self._device.readU8(BME280_REGISTER_TEMP_DATA + 1)
    xlsb = self._device.readU8(BME280_REGISTER_TEMP_DATA + 2)
    raw = ((msb << 16) | (lsb << 8) | xlsb) >> 4
    return raw

```

```

def read_raw_pressure(self):
    """Reads the raw (uncompensated) pressure level from the sensor."""
    """Assumes that the temperature has already been read """
    """i.e. that enough delay has been provided"""
    msb = self._device.readU8(BME280_REGISTER_PRESSURE_DATA)
    lsb = self._device.readU8(BME280_REGISTER_PRESSURE_DATA + 1)
    xlsb = self._device.readU8(BME280_REGISTER_PRESSURE_DATA + 2)
    raw = ((msb << 16) | (lsb << 8) | xlsb) >> 4
    return raw

def read_raw_humidity(self):
    """Assumes that the temperature has already been read """
    """i.e. that enough delay has been provided"""
    msb = self._device.readU8(BME280_REGISTER_HUMIDITY_DATA)
    lsb = self._device.readU8(BME280_REGISTER_HUMIDITY_DATA + 1)
    raw = (msb << 8) | lsb
    return raw

def read_temperature(self):
    """Get the compensated temperature in 0.01 of a degree celsius."""
    adc = self.read_raw_temp()
    var1 = ((adc >> 3) - (self.dig_T1 << 1)) * (self.dig_T2 >> 11)
    var2 = ((
        ((adc >> 4) - self.dig_T1) * ((adc >> 4) - self.dig_T1)) >> 12) *
        self.dig_T3) >> 14
    self.t_fine = var1 + var2
    return (self.t_fine * 5 + 128) >> 8

def read_pressure(self):
    """Gets the compensated pressure in Pascals."""
    adc = self.read_raw_pressure()
    var1 = self.t_fine - 128000
    var2 = var1 * var1 * self.dig_P6
    var2 = var2 + ((var1 * self.dig_P5) << 17)
    var2 = var2 + (self.dig_P4 << 35)
    var1 = (((var1 * var1 * self.dig_P3) >> 8) +
        ((var1 * self.dig_P2) >> 12))
    var1 = (((1 << 47) + var1) * self.dig_P1) >> 33

```

```

if var1 == 0:
    return 0
p = 1048576 - adc
p = (((p << 31) - var2) * 3125) // var1
var1 = (self.dig_P9 * (p >> 13) * (p >> 13)) >> 25
var2 = (self.dig_P8 * p) >> 19
return ((p + var1 + var2) >> 8) + (self.dig_P7 << 4)

def read_humidity(self):
    adc = self.read_raw_humidity()
    # print 'Raw humidity = {0:d}'.format (adc)
    h = self.t_fine - 76800
    h = ((((((adc << 14) - (self.dig_H4 << 20) - (self.dig_H5 * h)) +
        16384) >> 15) * ((((((h * self.dig_H6) >> 10) * ((h *
            self.dig_H3) >> 11) + 32768)) >> 10) + 2097152) *
            self.dig_H2 + 8192) >> 14))
    h = h - (((((h >> 15) * (h >> 15)) >> 7) * self.dig_H1) >> 4)
    h = 0 if h < 0 else h
    h = 419430400 if h > 419430400 else h
    return h >> 12

@property
def temperature(self):
    "Return the temperature in degrees."
    t = self.read_temperature()
    ti = t // 100
    td = t - ti * 100
    return "{:.{:02d}C".format(ti, td)

@property
def pressure(self):
    "Return the temperature in hPa."
    p = self.read_pressure() // 256
    pi = p // 100
    pd = p - pi * 100
    return "{:.{:02d}hPa".format(pi, pd)

@property

```

```

def humidity(self):
    "Return the humidity in percent."
    h = self.read_humidity()
    hi = h // 1024
    hd = h * 100 // 1024 - hi * 100
    return "{}.{:02d}%".format(hi, hd)

```

6.3.3. Programa para el envío de notificaciones vía *Twitter*

```

def tweet():
    # Verifica si hay una conexión a internet disponible
    wlan = network.WLAN(network.STA_IF)
    if not wlan.isconnected():
        print('No hay conexión a internet disponible no se
        puede mandar tuit')

        return
    # Configura tus credenciales de API de ThingTweet
    oled_ssd1306()
    api_key = '8T29HC3NDKBIJ4WP'
    message = 'LOS NIVELES DE CO2 SON DE RIESGO : {} Ppm, la temperatura
    es {} C,
    la humedad es {} y la Presión atmosférica es {}hPa'.format(CO2,
    temp, hum, pres)
    url = 'https://api.thingspeak.com/apps/thingtweet/1/statuses/update'
    headers = {'Content-Type': 'application/x-www-form-urlencoded'}
    data = 'api_key={}&status={}'.format(api_key, message)
    response = urequests.post(url, headers=headers, data=data)
    print(response.content)

```

Estancia de investigación en el extranjero

Se realizó una estancia en el *Centro MCIA (Motion Control and Industrial Applications)* de la *Universidad Politécnica de Cataluña*, *Terrasa, España* con una duración de 3 meses (marzo a mayo de 2023) bajo la supervisión del Dr. Alejandro Paredes Camacho, se trabajó en el tema

de investigación de la tesis, el principal objetivo fue realizar el ensamble de la tarjeta PCB y el desarrollo del análisis de los datos adquiridos.

Publicaciones y congresos

Se presentaron los siguientes artículos en congresos internacionales, que están relacionados con el procedimiento y el desarrollo del tema de tesis planteado en este documento:

- Artículo 1: Presentado en el *IV Congreso internacional de ciencias de la computación y XII Congreso nacional de ciencias de la computación*, en México, Puebla del 3 al 7 de Octubre del 2022, el cual será publicado en la Revista Research in Computing Science del IPN edición 2023.
- Artículo 2: Presentado en *el 2nd International Conference on Trends in Electronics and Health Informatics*, en México, Puebla del 7 al 9 de Diciembre del 2022.
- Artículo 3: En proceso de revisión para *International Journal of Combinatorial Optimization Problems and Informatics*.
- Publicación de capítulo de libro: Publicado en actas de la conferencia publicadas por *Springer Singapore Book Series: Lectures Notes in Networks and Systems (LNNS)*, edición 2023, <https://lc.cx/0129gq>
- Cartel: Presentación en el *XIII Congreso nacional de tecnología aplicada a ciencias de la salud & III Congreso internacional de tecnología aplicada a ciencias de la salud*, en Ciudad de México, México del 15 al 17 de Junio del 2023.

DESARROLLO DE UN SISTEMA EMBEBIDO PARA ADVERTIR SOBRE LAS CONDICIONES DE RIESGO DE CONTAGIO DE COVID-19 MEDIANTE EL MONITOREO DE LA CALIDAD DEL AIRE

Yair Romero López, Ricardo Álvarez González, Rodrigo Lucio Maya Ramírez,
Alba Maribel Sánchez Gálvez

Benemérita Universidad Autónoma de Puebla, Puebla, Pue. México
yair.romerolope@alumno.buap.mx, ricardo.alvarez@coreo.buap.mx,
rodrigo.maya@correo.buap.mx, alba.sanchez@correo.buap.mx

Resumen. El impacto súbito y generalizado de la pandemia del COVID-19, ha afectado a las actividades esenciales y presenciales de nuestra vida cotidiana, a pesar de la existencia de nuevas variantes del virus, existe un número significativo de la población vacunada en México, esto ha dado origen a la propuesta del retorno a las actividades presenciales, en sectores educativos, turísticos, profesionales, etc.

Por ello se requiere todas las medidas de protección e higiene, además de una planificación muy estricta, para minimizar el riesgo de contagio. La medición de la concentración de dióxido de carbono CO₂ es una estrategia que puede advertir el riesgo de contagio de la enfermedad del coronavirus (COVID-19) en un espacio cerrado donde se encuentre reunido un grupo de personas. El resultado puede proporcionar información, a partir de la cual se puede deducir si la ventilación es adecuada o deficiente, lo que facilitaría la propagación del virus. Esta es la razón por la cual se propone desarrollar un sistema embebido para monitorear la concentración de CO₂ en un espacio cerrado y emitir alarmas cuando se exceda el valor de 1000 ppm y adicionalmente generar un registro diario de los datos, almacenándolos en la nube para su posterior análisis.

Palabras clave: COVID-19, CO₂, Sistema embebido, IoT, Sensor.

1 Introducción

En los últimos años, la calidad del aire en interiores ha recibido una atención considerable de gobiernos ambientales, instituciones políticas y la comunidad de científicos internacionales, debido a su estrecha asociación con la salud pública, la comodidad y el bienestar de las personas [1][2].

Actualmente se puede hacer uso de las tecnologías emergentes para advertir sobre las condiciones de riesgo de contagio del COVID-19, el Internet de las cosas (IoT) muestra un gran potencial para medir datos en tiempo real de un ambiente cerrado que puedan auxiliar a los ocupantes de éstos espacios a tomar decisiones relevantes para

CO₂ MONITORING SYSTEM TO WARN OF POSSIBLE RISK OF SPREAD OF COVID 19 IN CLASSROOMS

Yair Romero López¹, Ricardo Álvarez González², Rodrigo Lucio Maya Ramírez³ Alba Maribel Sánchez Gálvez⁴

Benemérita Universidad Autónoma de Puebla
yair.romerolope@alumno.buap.mx¹, ricardo.alvarez@correo.buap.mx²,
rodrigo.maya@correo.buap.mx³, alba.sanchez@correo.buap.mx⁴

Abstract. The sudden and widespread impact of the COVID19 pandemic has affected the essential and face-to-face activities of our daily lives for almost two years; despite the existence of new variants of the virus, there are significant numbers of the population vaccinated internationally, this gave rise to the proposal to return to face-to-face activities, in the educational, tourist, professional sectors, etc. For this reason, all protection and hygiene measures are necessary, in addition to rigorous planning, to minimize the risk of contagion. Measuring the concentration of carbon dioxide (CO₂) is a strategy that can alert to the risk of contagion of the coronavirus disease (COVID-19) in an enclosed space where a group of people is gathered. The result can provide information from which one can deduce whether ventilation is adequate or poor, which would facilitate the spread of the virus. This is why it is proposed to develop an embedded system to monitor the concentration of CO₂ in a classroom and issue alarms when the value of 1000 ppm is exceeded. On the other hand, the activation of a device is done to maximize the flow of air, and thus the ventilation of the room is more efficient, in addition, a daily record of the data is carried out, storing them in the cloud for later analysis.

Keywords: Embedded system · CO₂ · IoT · COVID-19 · MQTT
· Sensor. · ThingSpeak

1 Introduction

In recent years, indoor air quality has received considerable attention from environmental governments, political institutions, and the international scientific community, due to its close association with public health, comfort, and well-being [1][2].

Open windows and a good heating, ventilation, and air conditioning (HVAC) system are starting points for keeping classrooms safe during the COVID-19 pandemic. But they are not the last word, according to a new study from researchers

CO₂ FORECASTING TO REDUCE INDOOR COVID-19 TRANSMISSION USING A TIME SERIES ALGORITHM

*Yair Romero López¹, Ricardo Álvarez González¹, Rodrigo Lucio Maya Ramírez¹,
Alba Maribel Sánchez Gálvez¹, Ismael Minor Sampedro¹, Alejandro Paredes Camacho²*

¹ Benemérita Universidad Autónoma de Puebla, Puebla, Pue. México.

² Dept. of Electronic Engineering, MCI-UPC Barcelona Tech, Terrassa, Spain.

yair.romerolope@alumno.buap.mx, ricardo.alvarez@correo.buap.mx,
rodrigo.maya@correo.buap.mx, alba.sanchez@correo.buap.mx, ismael.minor@alumno.buap.mx,
alejandro.paredes@upc.edu

Abstract. The impact generated because of COVID-19 pandemic has affected the face-to-face activities in our daily life, the existence of a considerable number of vaccinated people has given a proposal for the return of activities, on the other hand, people tend to spend most of their time in enclosed spaces causing a high concentration of carbon dioxide (CO₂). The measurement of CO₂ concentration is a strategy that can warn about the risk conditions of COVID-19 transmission indoors where a group of people is gathered. In this work, a method to measure CO₂ indoor and predict values for reducing the contagions of COVID-19 by means time series algorithm and the custom embedded system is presented. The application of this method during a defined period showed a series of high-precision PPM results.

Keywords: Indoor air quality, Embedded system, IoT, CO₂, COVID-19, ARIMA, Time series.

1 INTRODUCTION

Currently, indoor air quality (IAQ) has received considerable attention due to its close association with public health, comfort, and well-being [1][2]. A study by the Environmental Protection Agency (EPA) estimates that Americans spend 93% of their time indoors [3], resulting in high concentrations of CO₂, indoors with high numbers of people and reduced ventilation. Therefore, IAQ should be monitored due to a variety of health conditions it can cause, including COVID-19, which is the focus of this work. Recent studies have shown that high levels of CO₂, can lead to a higher incidence of COVID-19 infections among individuals congregating indoors [3][4].

The WHO suggests that COVID-19 is transmitted mainly through microdroplets. The inoculation of microdroplets in the respiratory tract results from the exposure of the host to patient events (coughing, sneezing, throat clearing, etc.) or procedures that induce the dispersion of droplets into the air. For the WHO, most infections occur through close contact, so the distance between people must be at least one meter and adequate indoor ventilation is essential [5][6].

Nowadays, emerging technologies can be used to warn about the risk conditions of contagion of COVID-19, the Internet of Things (IoT) shows great potential to measure data in real-time indoors, in this way It can help the occupants of these spaces to make decisions to improve indoor ventilation, mitigating the levels of CO₂, reducing the chances of contagion among the occupants. Since the conditions of CO₂ can worsen at any time depending on the environment in which the system is located, forecasting this data can be of great help to avoid scenarios where a massive contagion occurs and thus efficiently manage the tools for the improvement of air quality, taking preventive actions.

Even though there are prediction methods to reduce the spread of COVID-19 indoor and advanced embedded systems for measuring CO₂ levels, it is necessary to explore new methods forecast such as time series that reduce computing costs and low-cost embedded systems customized, low consumption, scalable and versatile.

In this work, a method for the prediction of CO₂ levels are presented to reduce the risks of contagion of

CO₂ Monitoring System to Warn of Possible Risk of Spread of COVID-19 in Classrooms



Yair Romero López, Ricardo Álvarez González,
Rodrigo Lucio Maya Ramírez, and Alba Maribel Sánchez Gálvez

1 **Abstract** The sudden and widespread impact of the COVID-19 pandemic has
2 affected the essential and face-to-face activities of our daily lives for almost two
3 years; despite the existence of new variants of the virus, there are significant num-
4 bers of the population vaccinated internationally, and this gave rise to the proposal
5 to return to face-to-face activities, in the educational, tourist, professional sectors,
6 etc. For this reason, all protection and hygiene measures are necessary, in addition to
7 rigorous planning, to minimize the risk of contagion. Measuring the concentration
8 of carbon dioxide (CO₂) is a strategy that can alert to the risk of contagion of the
9 coronavirus disease (COVID-19) in an enclosed space where a group of people is
10 gathered. The result can provide information from which one can deduce whether
11 ventilation is adequate or poor, which would facilitate the spread of the virus. This
12 is why it is proposed to develop an embedded system to monitor the concentration
13 of CO₂ in a classroom and issue alarms when the value of 1000 ppm is exceeded.
14 On the other hand, the activation of a device is done to maximize the flow of air, and
15 thus the ventilation of the room is more efficient; in addition, a daily record of the
16 data is carried out, storing them in the cloud for later analysis.

17 **Keywords** Embedded system · CO₂ · IoT · COVID-19 · MQTT · Sensor ·
18 ThingSpeak

Y. Romero López (✉) · R. Álvarez González · R. Lucio Maya Ramírez ·
A. Maribel Sánchez Gálvez
Benemérita Universidad Autónoma de Puebla, Puebla, Mexico
e-mail: yair.romerolope@alumno.buap.mx

R. Álvarez González
e-mail: ricardo.alvarez@correo.buap.mx

R. Lucio Maya Ramírez
e-mail: rodrigo.maya@correo.buap.mx

A. Maribel Sánchez Gálvez
e-mail: alba.sanchez@correo.buap.mx

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023
M. Mahmud et al. (eds.), *Proceedings of Trends in Electronics and Health
Informatics*, Lecture Notes in Networks and Systems 675,
https://doi.org/10.1007/978-981-99-1916-1_18

1

DESARROLLO DE UN SISTEMA EMPOTRADO IOT PARA ADVERTIR SOBRE LAS CONDICIONES DE LA CALIDAD DEL AIRE

Yair Romero López¹, Ricardo Alvarez Gonzalez¹, Rodrido Lucio Maya Ramirez¹, Alba Maribel Sanchez Galvez¹, Alejandro Paredes Camacho²

Maestría en Ingeniería Electrónica, BUAP, Puebla, México¹

Departamento de Ingeniería Electrónica, MCIA-UPC Barcelona Tech, Terrassa, España²



1. Introducción

Evaluar las condiciones de la calidad del aire es una estrategia que puede advertir el riesgo de contagio de la enfermedad de coronavirus o algunas otras enfermedades respiratorias, en un espacio cerrado donde se encuentre reunido un grupo de personas [1]. El resultado puede proporcionar información, a partir de la cual se puede deducir si la ventilación es adecuada o deficiente, lo que facilitaría la propagación del virus [2]; por lo cual se propone desarrollar un sistema empotrado con sensores y conexión IoT para monitorear las variables ambientales y la concentración de dióxido de carbono en un espacio cerrado. Si los niveles de CO₂ superan los parámetros establecidos de riesgo para la transmisión del virus, se activarán alarmas con grabaciones de voz y se aumentará la ventilación mediante un sistema eléctrico, posteriormente un análisis de datos nos ayudaría a obtener conclusiones valiosas, como la correlación de variables las cuales se usan para la climatización de espacios cerrados, lo cual se hace uso de las condiciones de temperatura, presión atmosférica y humedad relativa del lugar.

2. Uso de sensores CO₂

El uso de un sensor de CO₂ para controlar la calidad del aire puede disminuir el riesgo de contagio de enfermedades cardiovasculares y respiratorias, incluyendo el COVID-19 [3]. Según las asociaciones ASHRAE y REHVA, los niveles de CO₂ superiores a 1000 ppm en entornos interiores se han relacionado con diversos síntomas respiratorios y neurofisiológicos, como dolor de cabeza y fatiga. Esto sugiere que la falta de ventilación adecuada en un ambiente interior puede afectar negativamente a las personas que lo ocupan [4].

3. Objetivo

Se diseñará un sistema embebido capaz de realizar el sensado de los niveles de CO₂, temperatura, humedad relativa y presión atmosférica. Los datos recopilados serán enviados a una plataforma en línea para su posterior análisis y visualización en una interfaz gráfica de usuario. En caso de que el nivel de CO₂ supere el umbral de riesgo de 1000 ppm, el sistema enviará notificaciones de alerta tanto en forma de audio como en publicaciones en Twitter.

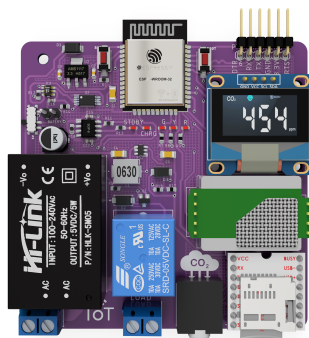


Figura 1: Diseño electrónico del sistema embebido

4. Metodología: Envío y recepción de datos

Se utilizará una Dashboard diseñada con Power BI data para recibir y almacenar datos obtenidos de un sistema embebido a través de un protocolo de comunicación inalámbrica MQTT. Los datos podrán ser descargados en formato .CSV y se presentarán visualmente en gráficos que muestran las mediciones de las variables monitoreadas organizados por fechas, así como la predicción de los próximos 10 datos, el promedio, el máximo, el mínimo y la ubicación del sistema embebido.

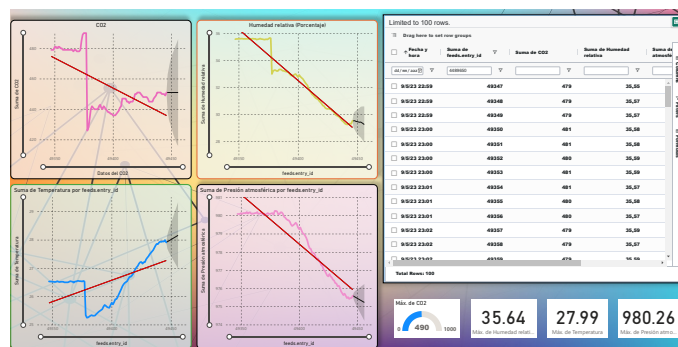


Figura 2: Datos visualizados en la dashboard

5. Resultados del análisis de datos

La relación entre las variables: temperatura, humedad relativa, presión atmosférica y concentración de CO₂ se determina con índice de correlación, lo que puede ayudar a comprender mejor los factores que influyen en la calidad del aire interior. Además, una vez que se ha encontrado una relación significativa, se pueden utilizar los valores de las variables ambientales para predecir la concentración futura de CO₂ [5].

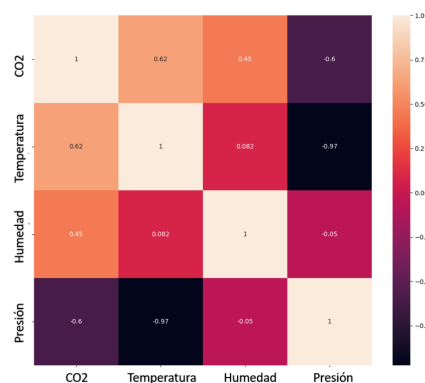


Figura 3: Matriz de correlación indicando la relación entre variables sensadas

7. Referencias

- [1] Gonçalo Marques Jagriti Saini, Maitreyee Dutta. Internet of things for indoor air quality monitoring. 25(2):197–206, 2021.
- [2] Jagriti Saini, Maitreyee Dutta, and Gonçalo Marques. Predicting Indoor Air Quality: Integrating IoT with Artificial Intelligence, pages 51–67. Springer International Publishing, Cham, 2021.
- [3] Organización Panamericana de la salud. Calidad del aire. <https://www.paho.org/es/temas/calidad-aire>, 2023.
- [4] National Collaborating Centre for Environmental Health. <https://www.nccch.ca>, 13 de Mayo de 2023.
- [5] Johanna Kallio, Jaakko Tervonen, Pauli Räsänen, Riku Mäkyinen, Jani Koivusaari, and Johannes Peltola. Forecasting office indoor CO₂ concentration using machine learning with a one-year dataset. *Building and Environment*, 187:107409, 2021.

6. Conclusiones

1. Con los resultados obtenidos se deduce que cuando hay una acumulación de CO₂ en un espacio cerrado, puede haber un aumento en la temperatura debido a que el CO₂ actúa como un gas de efecto invernadero, atrapando el calor en el ambiente. Además, el aumento del CO₂ puede disminuir la cantidad de oxígeno disponible en el aire, lo que puede provocar una sensación de bochorno y afectar la calidad del aire notablemente.
2. Es importante monitorear y mantener los niveles adecuados de CO₂, temperatura, humedad y presión atmosférica en los espacios cerrados para garantizar un ambiente saludable y seguro para las personas que los ocupan, en nuestro caso evitar un riesgo de contagio de COVID-19, sin embargo la extrapolación de este sistema puede ser conveniente para enfermedades respiratorias o asociadas a la mala calidad del aire, como lo son, neumonía, gripe estacional, etc.