



Universidad de  
Oviedo



# **ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.**

## **GRADO EN INGENIERÍA ELECTRÓNICA INDUSTRIAL Y AUTOMÁTICA**

### **DISEÑO Y FABRICACIÓN DE UN SENSOR DE CO<sub>2</sub>**

**D. Silva López, César**  
**TUTOR: D. Rubio García, Ramón**

**FECHA: Mayo, 2022**

# Memoria

# Índice

1. Introducción .....	7
1.1.- Motivación .....	7
1.2.- Objetivos y alcance.....	8
2. Estado del arte.....	10
2.1.- Dióxido de carbono (CO <sub>2</sub> ).....	10
2.2.- Tipos de sensores de CO <sub>2</sub> .....	11
2.2.1.- Sensores semiconductores de óxido metálico (MOS).....	12
2.2.2.- Sensores catalíticos .....	13
2.2.3.- Sensores electroquímicos .....	14
2.2.4.- Sensores NDIR .....	15
2.2.5.- Sensores foto acústicos .....	16
2.3.- Equipos de medición de CO <sub>2</sub> comerciales – Estado del mercado .....	17
2.4.- Investigaciones previas .....	21
3. Diseño e implementación .....	23
3.1.- Características técnicas y funcionales .....	23
3.2.- Composición básica del sistema.....	24
3.3.- Hardware - Elementos del sistema.....	25
3.3.1.- Sensor de CO <sub>2</sub> .....	25
3.3.2.- Zumbador .....	28
3.3.3.- Tira LED.....	29
3.3.4.- Microprocesador .....	31
3.3.5.- Carcasa – Impresión 3D.....	34
3.3.6.- Placa base – PCB.....	36
3.4.- Funcionamiento .....	41
3.4.1.- Consumo .....	46
3.5.- Software .....	47
3.5.1.- Programación ESP32 .....	47
3.5.2.- Base de datos .....	48
3.5.3.- Aplicación Web.....	50
4. Resultados experimentales .....	54
4.1.- Experimento en espacio MediaLab .....	54
4.2.- Experimento Aula Exámenes.....	56
4.3.- Resultados .....	58
5. Conclusiones y futuro .....	59
5.1.- Conclusiones.....	59

5.2.- Trabajos futuros.....	60
6. Referencias y bibliografía.....	61
Anexo I. Código fuente local.....	66
Anexo II. Código fuente Wifi.....	74
Anexo III. Gráficas.....	86



## Agradecimientos

*Tras redactar el documento que supone el fin de mis estudios en el grado de Ingeniería Electrónica Industrial y Automática, quisiera dejar patente mi agradecimiento dedicando unas breves líneas a todas las personas que, de una forma u otra, han sido de gran ayuda a la hora de cumplir mis objetivos de cara a obtener mi titulación.*

*En primer lugar, quiero agradecer a todo el equipo de MediaLab, un lugar que forma una familia encabezada por Ramón y donde he podido probarme y entrenar competencias que, aunque deberían formar parte de los estudios, no están presentes o no están reflejadas a diario en el discurso de la carrera. En especial, quiero mostrar mi agradecimiento a los componentes de mi equipo de trabajo; Juan Boto, Aida Álvarez e Iván Graña, pues sin ellos, de ninguna manera hubiera salido adelante todo este proyecto. Por primera vez en mi vida he trabajado en un espacio donde todos nos ayudamos a todos y estamos sometidos disfrutando continuamente de nuevas experiencias enriquecedoras que nos hacen crecer de forma profesional y personal. Gracias, Ramón.*

*Finalmente, no puedo olvidarme de todas aquellas personas con las que he compartido tiempo a lo largo del camino y que me han ayudado a superar los obstáculos y momentos de bajón que supone estudiar un grado como este y, por supuesto, a mi familia, que siempre ha tenido que sacrificarse para que yo pueda estar hoy aquí. Todo lo que diga de vosotros será siempre poco.*

# 1. Introducción

## 1.1.- Motivación

La pandemia sufrida en el año 2020 debido al SARS-CoV-2 puso de manifiesto lo vulnerable que es nuestra forma de vida; la globalización facilita, cada vez con más frecuencia, la movilidad y los contactos entre personas de diferentes lugares del mundo; esto hace que las poblaciones estén más expuestas a enfermedades fácilmente transmisibles. Esta situación pone en evidencia la falta de preparación y las dificultades de los países para afrontar los efectos de una enfermedad que se origina en un lugar determinado del mundo, y que pueda alcanzar el nivel de pandemia, poniendo el foco de atención en enfermedades que se pueden contagiar por vía aérea, como es el caso de la COVID19 [1].

No solo la globalización acentúa el riesgo de transmisión de todo tipo de virus. La frenética actividad social de nuestros días, los grandes eventos y las grandes aglomeraciones que se dan en la sociedad actual, constituyen un impedimento para la contención de patógenos que se transmiten de persona a persona. Tal es así, que en el año 2020 se llegó a una situación sin precedentes: confinamientos a nivel mundial, poniendo en jaque la economía y llegando a situaciones drásticas e insostenibles en muchas comunidades.

Sin embargo, el ser humano, como animal social, no puede quedarse aislado de por vida, por lo que hay que buscar la forma de tomar medidas y controlar las infecciones víricas debidas al SARS-CoV-2 (y otras infecciones similares), sin tener que renunciar a compartir espacios con otras personas. Una de esas formas es, controlar la calidad de aire en interiores, ya que hay estudios en los que se ha probado que la cantidad de CO<sub>2</sub> en un espacio cerrado tiene correlación con la facilidad de infección debido a virus que se transmiten por aerosoles [2].

Es ahí donde entra en juego este proyecto: construir un sensor de CO<sub>2</sub>, ya que como se ve en diferentes estudios y se ha comprobado con un experimento propio, el conocer los niveles de CO<sub>2</sub> en un espacio, hace que se pueda actuar sobre los sistemas de ventilación con el fin de reducirlos, facilitando la vuelta al trabajo presencial y pudiendo conocer la seguridad de los

espacios interiores de ocio, puesto que menores niveles de CO<sub>2</sub> implican mejor calidad del aire y mayor dificultad para infectarse con este tipo de virus [3].

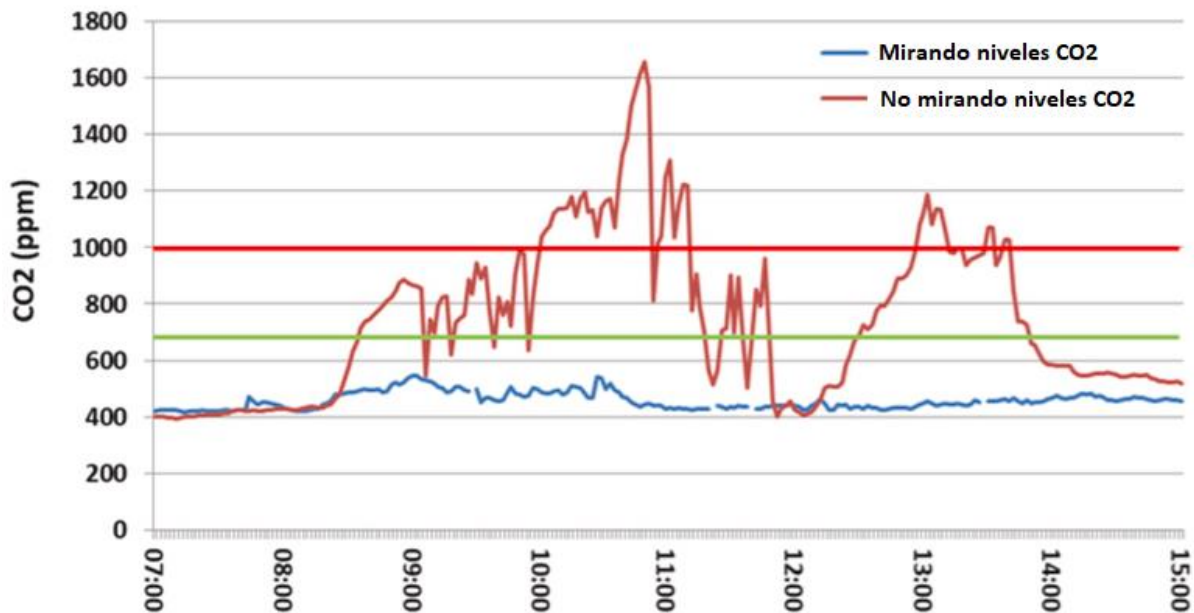


Figura 1.1.- Comparación de niveles de CO<sub>2</sub> mediante sensores, en una clase durante un experimento, teniendo en cuenta si se miran o no los niveles de CO<sub>2</sub> [3].

Por esta razón, este proyecto resulta muy interesante para aportar un granito de arena con la obtención de sensores precisos, baratos y transparentes, que puedan ser utilizados en muchos espacios; más ahora que, a fecha de redacción de este artículo, se están aplicando normativas para obligar a la hostelería a disponer de un medidor de CO<sub>2</sub> que controle los niveles del mismo en el interior de los locales para tener en cuenta cuando es necesario ventilar y así poder desarrollar su actividad con normalidad.[4].

## 1.2.- Objetivos y alcance

Antes de comenzar con el proyecto, resulta conveniente establecer unos objetivos claros, ya que, al tratarse de un campo relativamente nuevo, es necesario acotar el alcance de nuestra implementación, delimitando hasta donde se quiere llegar, sin cerrar la posibilidad de investigar posibles rutas de progreso o nuevo desarrollo para trabajos futuros.



Los objetivos que se perseguirán serán:

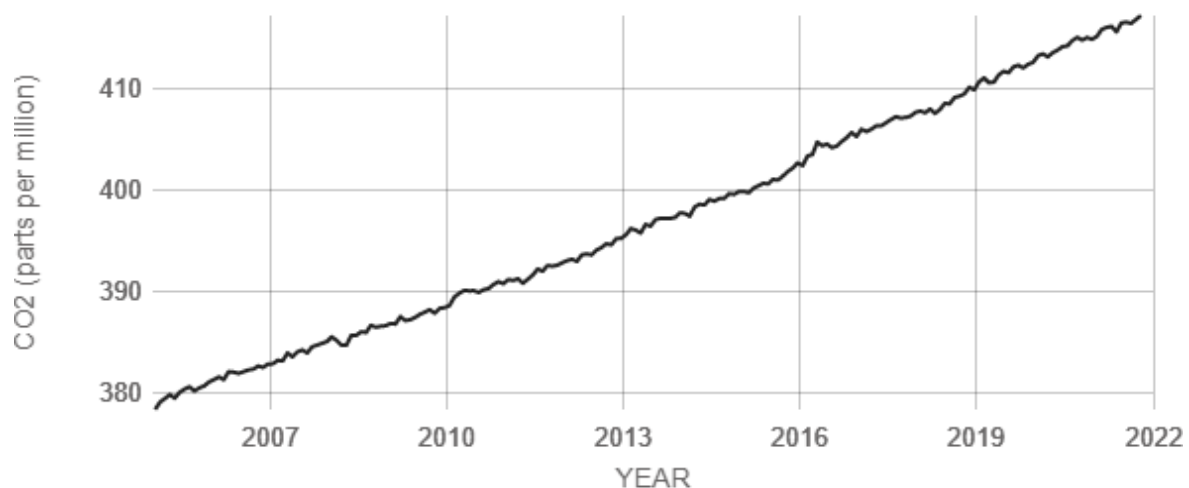
- Diseñar y fabricar un medidor de CO<sub>2</sub> para interiores de forma transparente, de manera que cualquiera pueda fabricar su sensor, en su casa, centro educativo, lugar de trabajo, etc. sin depender de medidores comerciales y sabiendo cómo funciona su sistema y las limitaciones que tiene.
- Intentar obtener la mayor precisión posible dentro de las limitaciones tecnológicas y de concepto implícito en el propio funcionamiento del sensor.
- Obtener las características de los mejores sensores comerciales del mercado, pero con un precio más reducido.
- Realizar pruebas de funcionamiento del sensor y comprobar que se cumplen las premisas para las que son diseñados (A más personas, más CO<sub>2</sub>. A más ventilación, mayor control de la concentración de CO<sub>2</sub>, etc.)
- Comprobar la evolución de la concentración de CO<sub>2</sub> en un espacio cerrado mediante el uso de los propios sensores.

## 2. Estado del arte

El paso previo a realizar el diseño será ofrecer una base teórica sobre aquello que se pretende medir y poner un contexto sobre el estado actual de la tecnología en el campo de la medición de CO<sub>2</sub> en el aire.

### 2.1.- Dióxido de carbono (CO<sub>2</sub>)

El dióxido de carbono es uno de los gases que componen la atmósfera terrestre. Es un gas de efecto invernadero con unas implicaciones muy concretas en el aumento de la temperatura global. Es bien sabido que la actividad humana desde la época de la industrialización ha hecho crecer la concentración de este gas hasta niveles sin precedentes. Actualmente, su concentración promedio en el aire ronda las 417 ppm, según análisis de la NASA[5].



Source: [climate.nasa.gov](https://climate.nasa.gov)

Figura 2.1.- Evolución temporal concentración de CO<sub>2</sub> en la atmósfera [5].

Este dato es interesante y uno de los más importantes puesto que, por lo que se verá más adelante, va a ser parte fundamental a la hora de establecer referencias y calibraciones en los sensores de CO<sub>2</sub>, tomando ese como el mínimo valor que se debería obtener en cualquier

ambiente de interiores (aunque se utilizará 400 ppm, ya que el dato de 417 ppm no deja de ser un promedio y varía dependiendo de la época del año y la localización física).

La parte que más interesa remarcar respecto del CO<sub>2</sub>, es la que tiene que ver con la respiración de los seres vivos, este proceso consiste en un intercambio de oxígeno y de CO<sub>2</sub>, se inhala el oxígeno y se espira el CO<sub>2</sub>. Teniendo en cuenta este proceso se puede sacar una relación entre la cantidad de CO<sub>2</sub> en un espacio cerrado y la probabilidad de que un virus sea más fácilmente transmisible, es decir, en espacios cerrados una mayor concentración de CO<sub>2</sub> significa una mayor exhalación de aire. lo que implica que, si hay algún individuo contagiado por algún tipo de virus, la probabilidad de contagio aumenta exponencialmente, sobre todo si el virus se transmite por aerosoles como es el caso del COVID-19.[6]

## 2.2.- Tipos de sensores de CO<sub>2</sub>

Actualmente existen en el mercado diferentes sensores de CO<sub>2</sub> que analizan la calidad del aire de diferentes formas. Este documento se referirá únicamente a las formas en las cuales se están comercializando diferentes sensores de CO<sub>2</sub>, si bien es cierto que puedan existir más formas que se están desarrollando en el momento o más métodos que no se utilizan de forma estandarizada para medir la calidad de aire en interiores.

En este estudio, cuando se habla de sensores de CO<sub>2</sub> se hace referencia a aquellos cuyo funcionamiento está pensado para interiores, con el objetivo de visualizar la información y actuar sobre ciertos parámetros de una sala o habitación para obtener una buena calidad de aire y cuya función es la de medir la concentración de CO<sub>2</sub> en dichos espacios en partes por millón (ppm). Quedarán fuera de esta ecuación aquellos métodos más genéricos de detección de gases o sensores específicos de uso industrial pensados para su uso en el control de procesos industriales químicos y de contaminación.

### 2.2.1.- Sensores semiconductores de óxido metálico (MOS)

Este tipo de sensores se fabrican utilizando la reacción de oxidación-reducción del gas en la superficie del semiconductor con el objetivo de cambiar el valor de resistencia del elemento sensible, y arrojando diferentes medidas dependiendo de este cambio de resistencia.

Este método de medida es en realidad un método no válido para medir con precisión la concentración de CO<sub>2</sub> en el aire, ya que este método no mide realmente la concentración de moléculas de CO<sub>2</sub> en el aire, sino que aproxima su concentración mediante una equivalencia a partir de una medición de VOC/TVOC (volatile organic compounds/total volatile organic compounds).

Los compuestos orgánicos son sustancias químicas que contienen carbono y se encuentran en todos los elementos vivos. Los compuestos orgánicos volátiles se convierten fácilmente en vapores o gases que contribuyen a la mala calidad del aire interior [7]. El problema es que al obtener la medida por aproximación a partir de un parámetro que depende de muchas variables, es complicado obtener una buena medición.

Las fuentes típicas de VOC en interiores incluyen agentes de limpieza, desinfectantes, ambientadores, deshumidificadores y toda una clase de sustancias que puedan encontrarse en interiores. Esto quiere decir que la medida de CO<sub>2</sub> equivalente que obtienen estos sensores, se aleja bastante del valor de concentración de CO<sub>2</sub> real, puesto que cualquier agente puede modificar la medida y arrojar valores que se encuentran lejos de los que se necesitan medir.

Se puede utilizar como ejemplo para esta categoría, el sensor CCS811 [8], utilizado en multitud de sensores comerciales y que se puede adquirir desde un precio tan bajo como 5€.



Figura 2.2.- Sensor de gases CCS811.

Se trata de un sensor de gases muy barato, de modestas características y cuyo uso para medir concentraciones de CO<sub>2</sub> es más bien limitado y no resulta demasiado interesante.

### 2.2.2.- Sensores catalíticos

Los sensores de tipo catalítico están formados por dos elementos: un elemento detector (D) que contiene material catalítico y es sensible al gas, y un elemento compensador (C) que es inerte. Los gases combustibles se quemarán únicamente en el elemento detector, provocando un aumento de su temperatura y, como consecuencia, un aumento de su resistencia. A cierta temperatura, el gas inflamable se quema catalíticamente en la superficie y ese es el principio del sensor de dióxido de carbono catalítico, también llamado de combustión térmica. En el compensador; su temperatura y resistencia permanecen sin cambios en presencia del gas [9].

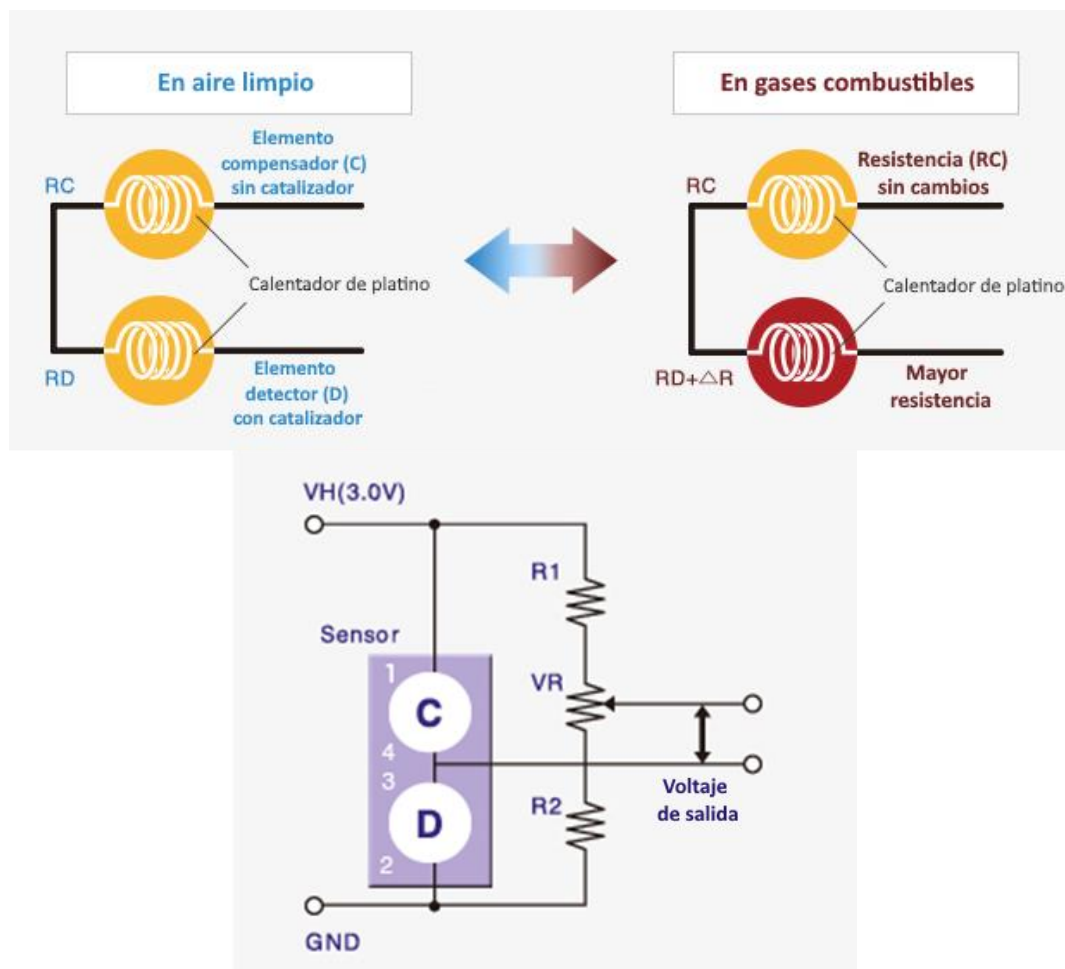


Figura 2.3.- Principio de funcionamiento de un sensor catalítico.

El circuito se forma como un puente de Wheatstone [10] con ambos elementos, como se muestra en la Figura 2.3. Se ajusta una resistencia variable (VR) para mantener un estado de equilibrio del circuito de puente en aire limpio y libre de gases combustibles.

En presencia de gases combustibles, sólo aumentará la resistencia del elemento detector, provocando un desequilibrio en el circuito del puente, produciendo así una señal de tensión de salida (Vout). La señal de voltaje de salida es proporcional a la concentración de gas que se quiere medir.

Este sensor ofrece unas medidas rápidas y relativamente precisas, además de tener una vida útil larga. Su principal problema, es que el rango de gases seleccionables es muy pequeño y puede llevar a medidas erróneas ante la presencia de otros gases que puedan ser quemados de igual forma. Además, no existen demasiadas opciones en el mercado y son difíciles de encontrar, por lo que se carece de un sensor de referencia que utilizar como ejemplo.

### 2.2.3.- Sensores electroquímicos

Los sensores electroquímicos de CO<sub>2</sub> son un tipo de sensor químico que funciona convirtiendo la presión parcial de la concentración de CO<sub>2</sub> en una señal eléctrica a través de una reacción electroquímica. Hay muchas formas en las que puede funcionar esta reacción electroquímica lo que lleva a que estos sensores pueden funcionar de muchas formas, como, por ejemplo, el sensor que utiliza Li<sub>7</sub>La<sub>3</sub>Zr<sub>2</sub>O<sub>12</sub> [11].

A nivel venta de sensores comerciales, no tienen mucho sentido debido a que sus desventajas (poca vida útil, problemas con la calibración, la medida se falsea por presencia de otros gases, etc.) superan con creces a las ventajas (simple y barato). De hecho, no se ha encontrado ningún sensor comercial de medida de CO<sub>2</sub> en interiores, fuera del ambiente industrial, por lo que se carece de un ejemplo claro que situar en esta categoría.

### 2.2.4.- Sensores NDIR

Esta categoría engloba a la gran mayoría de sensores de calidad que se utilizan para medir la calidad del aire en interiores. Se hace referencia a sensores mucho más complejos, caros y delicados. A cambio, se obtiene una vida útil larga (años), buena estabilidad, buena velocidad de respuesta y alta sensibilidad.

Actualmente, a fecha de redacción de este documento, es el único tipo de sensores que plantea la reciente legislación del uso obligatorio de sensores de CO<sub>2</sub> en espacios cerrados tales como gimnasios, restaurantes, aulas de clases, etc.[12]

A continuación, se va a resumir el principio de funcionamiento de los sensores NDIR [13].

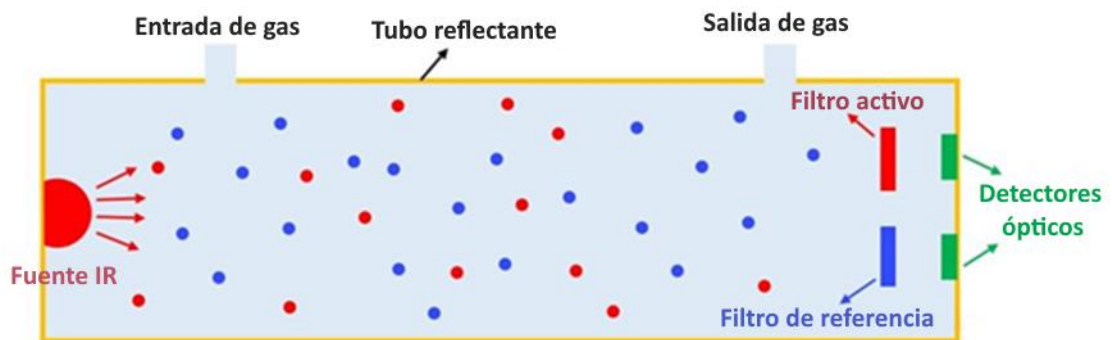


Figura 2.4.- Esquema funcionamiento sensor NDIR [13].

Una lámpara infrarroja (IR) hace brillar la luz a través de un tubo lleno de una muestra de aire y la dirige a un filtro colocado frente a un detector de luz infrarroja. El rango de radiación infrarroja producido por la lámpara es muy cercano al rango de absorción de dióxido de carbono. Cuando la luz infrarroja pasa a través del tubo, las moléculas de dióxido de carbono absorben un cierto rango de luz infrarroja mientras permiten el paso de otras longitudes de onda [14].

En el otro extremo, la luz restante llega al filtro óptico que absorbe todas las longitudes de onda de la luz excepto la longitud de onda absorbida por las moléculas de dióxido de carbono. El nivel de dióxido de carbono se obtiene midiendo la diferencia entre la luz emitida por una lámpara infrarroja y la cantidad de luz infrarroja recibida por el detector [15].

Un ejemplo de sensor NDIR es el Sensirion SCD30, uno de los sensores de más calidad del mercado y sobre el que se hablará en detalle más adelante.

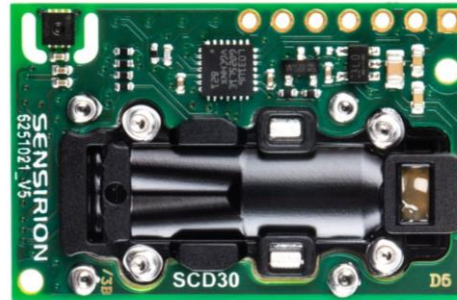


Figura 2.5.- Sensirion SCD30.

### 2.2.5.- Sensores foto acústicos

Esta es una nueva categoría de sensores en la que algunos fabricantes están dedicando parte de sus esfuerzos, debido a que tienen una buena precisión y sensibilidad (aunque no llega a cotas de los mejores NDIR) y se obtiene un producto compacto, durable y algo más barato de fabricar. Realmente su funcionamiento es muy similar a un NDIR con el añadido de que utiliza señales acústicas, pero se ha considerado utilizar una nueva categoría ya que podrían generalizarse más en el futuro reciente.

Como ejemplo, se va a utilizar el Sensirion SCD4x, cuyo esquema de funcionamiento se resume a continuación, cito textualmente la descripción general que ofrece el fabricante:

“La tecnología PASens® se basa en el principio de detección fotoacústica. El principio de detección fotoacústica permite una miniaturización extrema del sensor de CO<sub>2</sub> sin comprometer el rendimiento del sensor debido a que la sensibilidad del sensor es independiente del tamaño de la cavidad óptica. El tamaño reducido y la menor cantidad de componentes integrados dan como resultado una estructura de lista de materiales altamente rentable. Gracias a la miniaturización y la estructura económica de la lista de materiales de este nuevo sensor de CO<sub>2</sub>, el SCD4x abrirá numerosas posibilidades para un conjunto completamente nuevo de productos y aplicaciones.” [16]



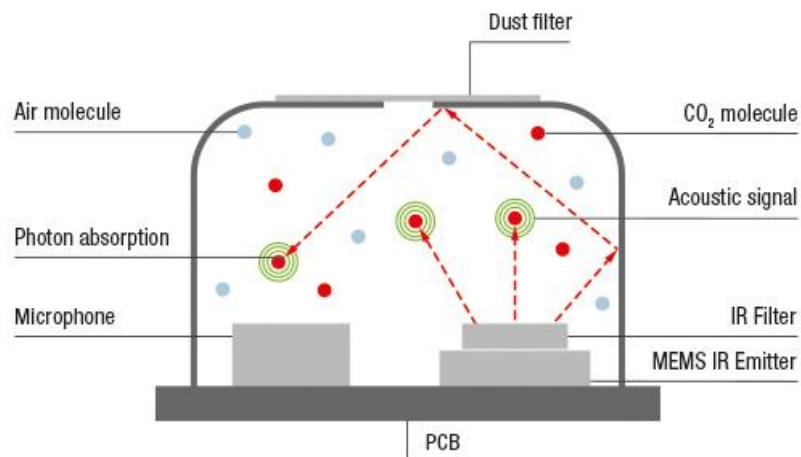


Figura 2.6.- Funcionamiento interno SCD4X [16].

### 2.3.- Equipos de medición de CO<sub>2</sub> comerciales – Estado del mercado

Debido al impulso comercial que han sufrido estos dispositivos con motivo de la pandemia y la obligatoriedad de su uso en determinados ambientes, muchos fabricantes se han lanzado a desarrollar sus propios medidores de CO<sub>2</sub>, que normalmente comparten características que permiten la visualización de la concentración de CO<sub>2</sub> en ppm, temperatura y humedad.

En la siguiente tabla se señalan unas pinceladas a modo de ejemplo de los sensores que se ofrecen en el mercado, indicando sus características y precio. Se mostrarán de menor a mayor coste.

Medidor	Tecnología sensor	Sensor	Características extra	Precio	link
EasyFire Air quality monitor	MOS (TVOC)	No definido	Muestra de información, batería, alerta sonora	47,99€	[17] Amazon
TACKLY CO <sub>2</sub> detector	NDIR	MH-Z19 (15€)	Muestra información, batería, alerta sonora	69,99€	[18] Amazon
Dioxcare	NDIR	SenseAir S8 (25€-30€)	Muestra información, batería, alerta sonora	139€	[19] Web Oficial fabricante
AirGuardian Mini	NDIR	No definido	Muestra información, batería, alerta sonora, acceso a aplicación (conexión)	265€	[20] Web oficial fabricante

Tabla 2.1.- Comparación de diferentes sensores comerciales.

Los precios se han tenido en cuenta a fecha de redacción de este artículo. Según muestran los datos, se puede comprobar que cuando aumenta el precio, mejoran las características tanto del sensor como las características de la estación de medición en su totalidad.

También se puede observar como el incremento de precio puede llevar a encontrar artículos relativamente caros, lo que hace que las personas tiendan a optar por perder características con el fin de obtener un precio más equilibrado, como se puede ver en un sensor superventas de Amazon como es el TACKLY, mencionado en la tabla. Este sensor utiliza un sensor de calidad inferior a cambio de un precio más competitivo.

Disponer de una unidad de este sensor ha sido de gran utilidad para conocer el tipo de sensor que lleva y descubrir donde se han hecho recortes para obtener un precio más contenido. La idea es que el sensor diseñado no exceda por mucho el precio de este, pero se pueda obtener

un amplio abanico de características extra que hagan interesante su uso por encima de estos superventas.



Figura 2.7.- Sensor CO<sub>2</sub> TACKLY.



Figura 2.8.- Despiece del sensor CO<sub>2</sub> TACKLY.

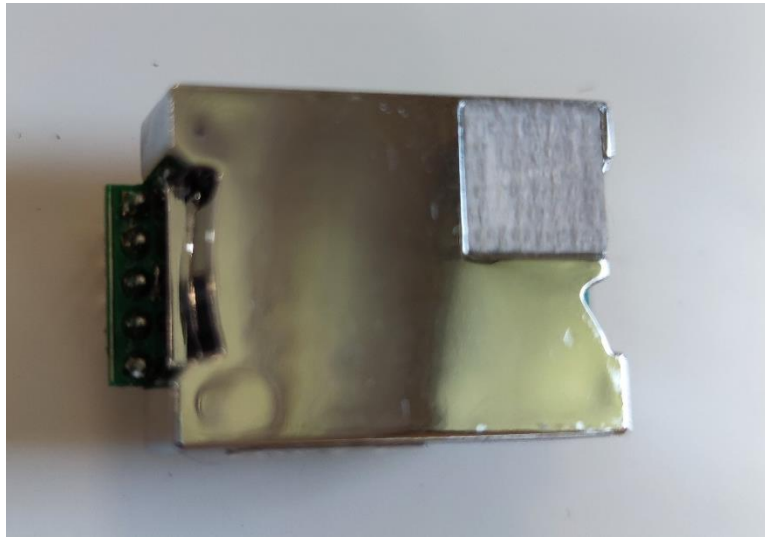


Figura 2.8.- Sensor MH-Z19 del sensor CO<sub>2</sub> TACKLY.

Al haber realizado una ingeniería inversa de este modelo de medidos de CO<sub>2</sub>, se puede apreciar claramente en que aspectos se ha recortado para mantener un precio competitivo. En primer lugar, el sensor utilizado es un sensor MH-Z19 [21], en cuya hoja de características podemos observar que su precisión es de 50 ppm + 5% del valor en ppm de la medida, lo que sumado al error de repetibilidad (el cual no especifica el fabricante), tenemos que para valores altos de concentración de CO<sub>2</sub>, el error se dispara. Se trata de un sensor con la calidad justa, pero de muy bajo precio, con comunicación vía UART o PWM, que no resulta la mejor opción para este tipo de dispositivos.

Por otro lado, carece de un microprocesador con potencia suficiente para manejar más instrucciones que la simple toma y representación de datos en los leds de los que dispone, con ausencias como la de poder conectarse a una red wifi. Sin embargo, como puntos positivos, este sensor plantea un diseño sencillo y visualmente atractivo.

## 2.4.- Investigaciones previas

El crecimiento en ventas de estos dispositivos junto con la pandemia sufrida debido al SARS-CoV-2, ha favorecido un área nueva de investigación para contener la emergencia médica y eso implica un crecimiento en la información del funcionamiento del CO<sub>2</sub> en interiores y su correlación con la cantidad de partículas víricas en un ambiente determinado.

Por ello, han surgido un gran número de estudios que arrojan luz y que servirán como base y fuente de información para seguir estudiando el tema o utilizar los datos obtenidos con fines experimentales o de comprobación.

En paso previo al diseño de los sensores y para tratar de alcanzar los objetivos de forma experimental, se ha realizado una búsqueda exhaustiva de estudios en el área de la relación CO<sub>2</sub>-Covid-Ventilación, para crear un marco previo de investigación en las que se ponen a prueba dichas relaciones y donde se obtiene información muy interesante al respecto. Muchos de los datos recogidos de estos trabajos, se utilizarán como bibliografía general para poder realizar ciertas afirmaciones sobre cómo se expande el virus y cómo se realizan ventilaciones correctamente.

Los estudios que se han consultado como investigación previa, además de los citados con anterioridad durante el transcurso de este documento, han sido los siguientes: [22]–[37].

Las conclusiones que se pueden obtener de los citados estudios en su conjunto son:

- Mayor número de personas implica mayores concentraciones de CO<sub>2</sub>. El hecho de tener más personas en un mismo espacio implica mayor probabilidad de contagio, por lo que se puede demostrar que a mayores concentraciones de CO<sub>2</sub>, hay una mayor probabilidad de contagio por aerosoles.
- La ventilación, tanto natural como forzada, son un factor clave en la contención de contagios, debido a que, como se ha demostrado, ventilar implica reducir los valores de CO<sub>2</sub>, lo que tiene como consecuencia una reducción en las probabilidades de contagio.
- Técnicas de ventilación más avanzada, como la ventilación cruzada, favorece flujos de corriente mejor pensados y que conllevan valores más bajos de CO<sub>2</sub>

- Un buen control de la concentración de CO<sub>2</sub> en un espacio cerrado es compatible con la vuelta a actividades de trabajo y de ocio que obliguen necesariamente a compartir espacios cerrados con concentraciones habituales de personas.
- Una buena ventilación implica de forma secundaria ahorro energético y estabilización de las temperaturas en épocas calurosas.
- El tener acceso a información de la concentración de CO<sub>2</sub> implica una actuación mejor sobre los elementos de ventilación, garantizando ahorro energético, temperaturas más agradables y la posibilidad de un control óptimo.
- Valores más bajos de CO<sub>2</sub> no solo implican menor posibilidad de contagio, si no otros beneficios sobre la salud que implican un ambiente más sano y mejor rendimiento de las personas que en ambientes con concentraciones excesivas de CO<sub>2</sub>.
- El uso de sensores de CO<sub>2</sub> contribuyen eficazmente a conseguir ambientes sanos y con un riesgo más bajo de transmisión de patógenos.

## 3. Diseño e implementación

Habiendo establecido un marco teórico previo y un primer vistazo a la situación actual del mercado en cuanto a técnicas de medición de CO<sub>2</sub>, y una vez investigado el tema, se puede plantear un diseño propio para un equipo de medición de CO<sub>2</sub>.

El diseño se llevará a cabo de forma que el equipo de medición de CO<sub>2</sub> propuesto se divida en dos partes diferenciadas: diseño de la parte hardware y diseño de la parte software.

### 3.1.- Características técnicas y funcionales

El equipo de medición de CO<sub>2</sub> que aquí se expone y se diseña, contará con las siguientes características:

- Medición de CO<sub>2</sub>, temperatura y humedad.
- Alta precisión y fiabilidad.
- Durabilidad y poca deriva temporal.
- Código de colores para visualizar el nivel de CO<sub>2</sub> de forma sencilla.
- Alarma sonora.
- Conexión Wifi.
- Conexión a base de datos.
- Conexión a una aplicación web para la visualización ordenada de datos, entre otras cosas.

Con vistas a facilitar la comprensión de los elementos utilizados y la propuesta de diseño, se va a simplificar la idea del funcionamiento esperado para el medidor de CO<sub>2</sub>. Más adelante se detallará todo este funcionamiento.

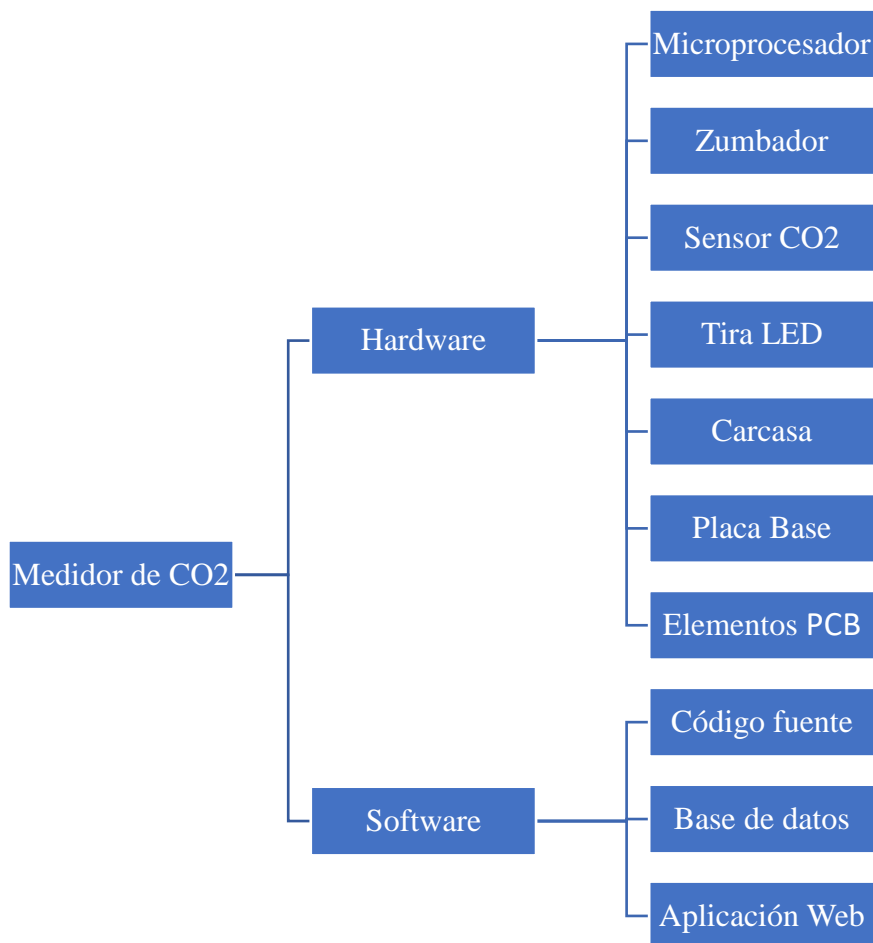
La idea del diseño es que visualmente sea similar a un semáforo, lo que facilitará su comprensión para todo el mundo, incluidos niños pequeños. Dispondrá de tiras LED, colocadas en 3 filas de 6 leds, que se iluminarán en verde cuando la calidad del aire sea buena (<700 ppm

de CO<sub>2</sub>), ámbar cuando la calidad del aire comience a empeorar (700 ppm-1000 ppm) y rojo cuando la calidad del aire sea mala y urja ventilar (>1000 ppm). Además, contará con una alarma sonora que emita un pitido cuando la calidad del aire pase a un nivel de calidad inferior (un pitido de verde a ámbar y dos pitidos de ámbar a rojo).

El medidor de CO<sub>2</sub> recogerá datos y mediante conexión vía Wifi los enviará a una base de datos donde se podrán visualizar en una aplicación web, así como históricos, máximos de interés y alertas.

### 3.2.- Composición básica del sistema

En este apartado se realizará un esquema básico y general de cómo está compuesto el sistema diseñado, separando los diferentes sistemas que componen el medidor de CO<sub>2</sub>.



Esquema 3.1.- Elementos que componen el diseño del medidor de CO<sub>2</sub>.



### 3.3.- Hardware - Elementos del sistema

En primer lugar, se procederá a exponer todo lo que tiene que ver con la construcción de la parte física del equipo de medición de CO<sub>2</sub>, de forma que quede completamente resuelto el apartado hardware y el funcionamiento de éste. El presupuesto que comprende todos los elementos se describirá en el documento “Presupuesto”.

#### 3.3.1.- Sensor de CO<sub>2</sub>

Debido a que en esencia este proyecto consiste en medir el CO<sub>2</sub>, se está hablando de la parte más crítica del sistema. Se necesita un sensor que ofrezca las características de más alta gama del mercado, con el menor error de medición posible para asegurar que los resultados de la medición sean los más precisos que se puedan obtener.

Para considerarse un sensor preciso, vamos a demandar que tenga la precisión de los mejores del mercado y para ello se establecerá el objetivo de obtener una precisión de  $\pm 70$  ppm cuando la medida sea de 1000 ppm, concentración que representaría el último cambio en las indicaciones de ventilación del medidor. Es sabido por las datasheet de los diferentes fabricantes que el error de medida depende directamente de la concentración de CO<sub>2</sub> medida. Se sabe de igual forma que la temperatura afecta a la precisión de la medida, por lo que se busca un sensor que tenga algún algoritmo interno que compense esa deriva debida a la temperatura.

Para esta función se ha elegido el sensor SCD30, del fabricante suizo de sensores “Sensirion”. Se trata de un producto “high-end” con características punteras, capaz de arrojar medidas de concentración de CO<sub>2</sub> en el aire con un bajo error, además de medir temperatura y humedad y poder utilizar esas medidas para establecer correcciones de ppm de CO<sub>2</sub> a medida que estos cambian. El error máximo del SCD30, de acuerdo con lo descrito en la tabla 3.1 (pág. 24) extraída de la datasheet, sería:

$$\text{Error (ppm)} = \text{Error base} + \text{Error \% medida} \times \text{Medida} + \text{Error repitibilidad}$$

Para medidas de 500 ppm:

$$Error_{500ppm} = 30 + 0,03 \times 500 + 10 = 55 \text{ ppm} \quad (3.1)$$

Para medidas de 1000 ppm:

$$Error_{1000ppm} = 30 + 0,03 \times 1000 + 10 = 70 \text{ ppm} \quad (3.2)$$



Figura 3.1.- Sensor CO<sub>2</sub> Sensirion SCD30.

En la tabla se pueden ver las características del sensor, extraídas directamente de la datasheet elaborada por el fabricante [38], donde se observa el alto grado de precisión, cuyas tolerancias pasan a ser de 30 ppm base. Por otro lado, detallar que cuenta con interfaz I<sup>2</sup>C, lo cual resulta una ventaja debido a su facilidad de uso.

Para llevar a cabo la calibración del sensor, se han calibrado manualmente en el aire exterior tras un periodo de adaptación de una semana, otorgándole el valor de 417 ppm, siguiendo las pautas establecidas por el fabricante [39]. El SCD30 dispone a su vez de un sistema de recalibrado automático que quedará activado. Este sistema funciona de tal forma que, durante una semana, analiza el comportamiento del CO<sub>2</sub> y, a partir de ese momento, toma el valor mínimo de CO<sub>2</sub> encontrado y lo sitúa en 400 ppm (ya que, como se comentó anteriormente, no puede haber menos de esa concentración de CO<sub>2</sub> en el aire), lo que servirá como base para el resto de medidas.

## 1 Sensor Specifications<sup>1</sup>

### CO<sub>2</sub> Sensor Specifications

Parameter	Conditions	Value
CO <sub>2</sub> measurement range	I2C, UART PWM	0 – 40'000 ppm 0 – 5'000 ppm
Accuracy <sup>2</sup>	400 ppm – 10'000 ppm	± (30 ppm + 3%MV)
Repeatability <sup>3</sup>	400 ppm – 10'000 ppm	± 10 ppm
Temperature stability <sup>4</sup>	T = 0 ... 50°C	± 2.5 ppm / °C
Response time <sup>5</sup>	$\tau_{63\%}$	20 s
Accuracy drift over lifetime <sup>6</sup>	400 ppm – 10'000 ppm ASC field-calibration algorithm activated and SCD30 in environment allowing for ASC, or FRC field-calibration algorithm applied.	± 50 ppm

Table 1: SCD30 CO<sub>2</sub> sensor specifications

### Humidity Sensor Specifications<sup>7</sup>

Parameter	Conditions	Value
Humidity measurement range	-	0 %RH – 100 %RH
Accuracy <sup>8</sup>	25°C, 0 – 100 %RH	± 3 %RH
Repeatability <sup>3</sup>	-	± 0.1 %RH
Response time <sup>5</sup>	$\tau_{63\%}$	8 s
Accuracy drift	-	< 0.25 %RH / year

Table 2: SCD30 humidity sensor specifications

### Temperature Sensor Specifications<sup>7</sup>

Parameter	Conditions	Value
Temperature measurement range <sup>9</sup>	-	- 40°C – 70°C
Accuracy <sup>8</sup>	0 – 50°C	± (0.4°C + 0.023 × (T [°C] – 25°C))
Repeatability <sup>3</sup>	-	± 0.1°C
Response time <sup>5</sup>	$\tau_{63\%}$	> 10 s
Accuracy drift	-	< 0.03 °C / year

Table 3: SCD30 temperature sensor specifications

Tabla 3.1.-Especificaciones del sensor SCD30 [38].

Respecto al consumo, de la misma hoja de datos podemos obtener los consumos, que son de 19 mA cuando funciona en forma continua, a los que hay que añadirle los 75 mA, necesarios para cuando se toma la medida. Con esto, tenemos un consumo de corriente de 94 mA. Teniendo en cuenta que el SCD30 puede funcionar con 3,3 V, la máxima potencia demandada por el dispositivo será:

$$P_{max_{SCD30}} = V \times I = 3,3 V \times 0,094 A = 0,31 W \quad (3.3)$$

### 3.3.2.- Zumbador

Con el fin de producir una alerta sonora, se utilizará un zumbador eléctrico. Entre sus características vamos a buscar que sea de tipo pasivo, ya que de esta forma podremos escapar de las limitaciones de los zumbadores activos, al carecer estos de posibilidad de variar el tono. Se utilizará un zumbador barato, que no encarezca el precio, que funcione a 5V o 3,3V, por ser las interfaces más comunes y que sea del tipo pasivo, con el fin de obtener un pitido notable pero agradable para el oído humano.

El módulo de alarma de altavoz KY-006, es un zumbador pasivo que resulta perfecto para este cometido, debido a su facilidad de uso y su bajo coste. Consiste en un botón piezoeléctrico pasivo que puede generar tonos entre 1,5 kHz y 2,5 kHz utilizando una señal cuadrada [40].

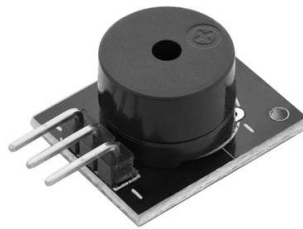


Figura 3.2.- Buzzer KY-006.

Este módulo estará conectado al microcontrolador con la idea de poder gestionar cuando se activa y la frecuencia de sonido que emitirá.

Este dispositivo funciona conectado a 5 V y tiene un consumo de corriente que siempre es inferior a 25 mA [41]. Además, hay que tener en cuenta que solo funciona durante un breve lapso. La potencia máxima demandada por este dispositivo será:

$$P_{max_{KY006}} = V \times I = 5 V \times 0,025 A = 0,125 W \quad (3.4)$$

### 3.3.3.- Tira LED

Para visualizar los diferentes colores de indicación de concentración de CO<sub>2</sub>, necesitamos una tira de leds RGB con la posibilidad de que cada led individual pueda mostrar diferentes colores (en este caso; verde, ámbar y rojo). Siguiendo las mismas exigencias que para el zumbador, se necesitará que puedan funcionar a 5 V o 3,3 V.

Con el objetivo de obtener un foco de luz bastante constante, se buscará disponer de 3 a 6 leds por cada 10 cm, lo que supone elegir entre tiras que vayan de los 30 leds/m a los 60 leds/m. Más de 60 leds/m tendría un impacto negativo en los consumos y posiblemente se necesitará una fuente de alimentación externa para los leds, de lo cual se prefiere prescindir. Menos de 30 leds/m haría que por cada fila solo tuviéramos uno o dos leds, lo cual resulta una iluminación presumiblemente escasa para este cometido.

Para la ejecución de este proyecto, la tira contará con LED elegida será la WS2812B [42] que permite variar el color aplicándole el código RGB con facilidad. Funciona con 5V y dispone de 6 leds por cada 10 centímetros, lo que aporta un foco de luz suficientemente bueno para una buena visibilidad. También nos permite encender solo 3 leds por fila en el caso de ser necesario ahorrar en consumo.

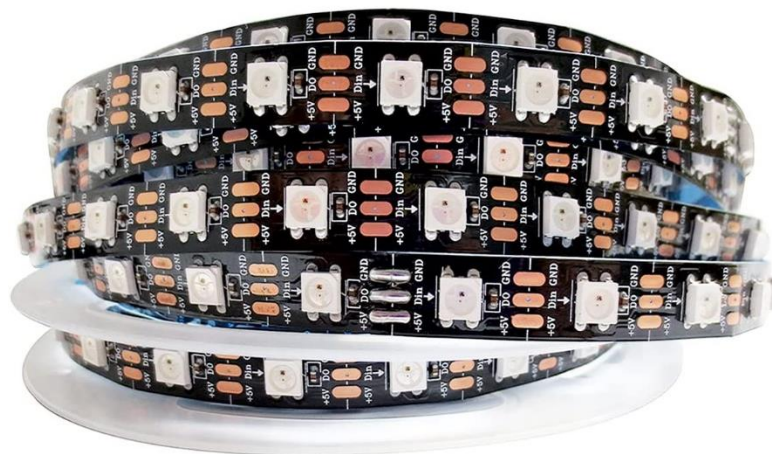


Figura 3.3.- Tira de Leds.

Hay que tener en cuenta que para la señal de control se necesitará una resistencia de limitación, ya que necesita de muy poca corriente para funcionar. En la datasheet [42] se puede

apreciar que se recomienda utilizar una resistencia de entre 300  $\Omega$  y 500  $\Omega$ , por lo que se ha optado por una resistencia normalizada de 470  $\Omega$  con una tolerancia del 5% de las más económicas. Su valor oscila entre 446,5  $\Omega$  y 493,5  $\Omega$ , resultando adecuada para la aplicación según los datos del fabricante del controlador RGB.

A continuación, se estudiará la demanda energética en el caso de la tira de leds. Hay que tener en cuenta que como máximo se encenderán 6 leds simultáneamente. Dentro de las características de la tira utilizada, podemos apreciar la tensión que utiliza para mostrar cada color, así como la corriente que necesita por componente, que resulta ser de 20 mA. Si se quisiese iluminar los 6 leds con luz blanca a toda potencia, necesitaríamos 60 mA para cada uno de los leds, pero en este caso se utilizarán otros colores y se situará la potencia lumínica al 40% de su capacidad, lo cual nos ofrece un buen balance entre visibilidad y consumo energético.

- Caso 1: Leds encendidos en verde. Para mostrar el verde, se utiliza un componente (G,255) que demanda como máximo 20 mA y necesita 3,4 V de tensión.

$$\text{Verde} \rightarrow 3,4 V \times 20 mA = 0,068 W / \text{unidad} \quad (3.5)$$

$$0,068 W / \text{unidad} \times 6 \text{ unidades} = 0,408 W \quad (3.6)$$

$$0,408 W \times 0,4 = 0,163 W \quad (3.7)$$

- Caso 2: Leds encendidos en ámbar. Para mostrar ámbar, se utilizan dos componentes (G,165 y R,255) que demanda como máximo 40 mA y necesita 2,4 V de tensión.

$$\text{Ámbar} \rightarrow 2,4 V \times 40 mA = 0,096 W / \text{unidad} \quad (3.8)$$

$$0,096 W / \text{unidad} \times 6 \text{ unidades} = 0,576 W \quad (3.9)$$

$$0,576 W \times 0,4 = 0,23 W \quad (3.10)$$

- Caso 3: Leds encendidos en rojo. Para mostrar el rojo, se utiliza un componente (R,255) que demanda como máximo 20 mA y necesita 2,2 V de tensión.

$$\text{Rojo} \rightarrow 2,2 V \times 20 mA = 0,044 W / \text{unidad} \quad (3.11)$$

$$0,044 W / \text{unidad} \times 6 \text{ unidades} = 0,264 W \quad (3.12)$$

$$0,264 W \times 0,4 = 0,1 W \quad (3.13)$$

Como se puede apreciar, el caso de mayor potencia demandada será cuando los leds se iluminen en color ámbar, ya que al utilizar dos componentes RGB, se necesitará un máximo de 40 mA para mostrar el color, llevando a un consumo de corriente máximo de 240 mA a plena potencia lumínica, que aplicando el factor del 40% de potencia supone un consumo de 96 mA.

### 3.3.4.- Microprocesador

El microprocesador constituye una parte fundamental del sistema y se debe encontrar un modelo que pueda satisfacer las necesidades de diseño requeridas por el resto de los componentes. Este elemento será el encargado de gestionar y controlar las medidas del sensor de CO<sub>2</sub>, enviar datos mediante conexión Wifi, enviar señales de control para las alertas sonoras y los cambios de color de la tira led. A continuación, se citarán las exigencias que se debe pedir al microprocesador para cumplir las exigencias del proyecto:

- Capacidad de ofrecer tensiones de 5 V (leds y zumbador) y 3,3 V (sensor de CO<sub>2</sub>) simultáneamente.
- Capacidad de ofrecer la corriente necesaria para abastecer todos los componentes. En el caso más desfavorable:

$$I_{max} = 96 mA (\text{leds}) + 25 mA (\text{zumbador}) + 94 mA (\text{sensor CO}_2) = 215 mA \quad (3.13)$$

- Capacidad de conectarse a redes Wifi.

- Capacidad de ser programado con lenguajes de alto nivel, facilitando la tarea de programado del micro.
- Capacidad de utilizar la interfaz I<sup>2</sup>C.
- Dos o más salidas digitales para enviar las señales de control de los leds y el zumbador.
- Potencia de computación suficiente para ejecutar un código de forma fluida. Se buscará que disponga al menos de un núcleo a una frecuencia de 200 MHz y 256 kB de memoria RAM.
- 1,5 MB o más de memoria ROM para alojar el programa, que ocupa unos 0,92 MB en su versión más extensa.

```
El Sketch usa 919354 bytes (70%) del espacio de almacenamiento de programa.  
Las variables Globales usan 40200 bytes (12%) de la memoria dinámica, dejando
```

Figura 3.4.- Espacio que ocupa el programa una vez compilado.

Para esta ejecución del proyecto, el microprocesador elegido es, en realidad, una placa de desarrollo ESP32 del fabricante AZ-Delivery [43], basada en el ESP32 de Espressif, que cuenta con las siguientes características que son de interés y pueden ser extraídas de la datasheet oficial [44]:



Figura 3.5.- ESP32 AZ-Delivery.

- Microprocesador WROOM-32 dual core a 240 MHz con Wifi 2.4Ghz + Bluetooth + 4 Mb de memoria programa y 512 kB de RAM, lo cual resulta interesante para disponer de una conectividad avanzada dentro de una misma placa a un precio contenido.
- Ofrece dos pines con diferentes tensiones de 5 V y 3,3 V, ofreciendo sin problema las tensiones que ofrece siendo alimentado vía micro USB.



- Ofrece corrientes de hasta 1200 mA para alimentar los diferentes componentes, siempre y cuando alimentemos el sistema con un adaptador de corriente lo suficientemente potente.

Table 13: Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	-0.3	3.6	V
$I_{output}^*$	Cumulative IO output current	-	1200	mA
$T_{store}$	Storage temperature	-40	150	°C

\* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3\_RTC, VDD3P3\_CPU, VDD\_SDIO) output high logic level to ground.

Tabla 3.2.- Parámetros del ESP32 [44].

- Diferentes pines de entrada y salida analógicos y digitales, convertor analógico-digital.
- Interfaz I<sup>2</sup>C, que será utilizada para establecer la comunicación entre el microprocesador y el sensor de CO<sub>2</sub>.
- Interfaz SPI, UART y PWM, útiles de cara a poder utilizar otros sensores o servicios en el futuro.
- Posibilidad de programación en Arduino.

Una vez descritas las bondades, se van a destacar unas indicaciones respecto del ESP32. En primer lugar, en relación con su consumo. Se sabe que además de los 215 mA de corriente máxima demandada por los periféricos, el propio micro necesita ser alimentado. De esta forma, en la datasheet podemos ver su consumo en modo activo, lo que pone en funcionamiento todos los módulos del ESP32. Con el wifi activo, se pueden ver consumos típicos de 240 mA. Para caminar del lado de la prudencia, se va a suponer que el ESP32 pueda alcanzar consumos de 500 - 600 mA, ya que, según el propio fabricante, si se va a alimentar por micro USB, se recomienda una corriente mínima de 600 mA para la alimentación de la placa.

Con todo esto, se tiene que el consumo máximo de corriente del sistema será de 600 mA + 215 mA = 815 mA, por lo que se puede concluir que necesitamos un adaptador de corriente que al menos ofrezca 1 A y 5 V para alimentar este sistema vía micro USB.

Table 17: RF Power-Consumption Specifications

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

Tabla 3.3.- Consumo de corriente del ESP32 en modo activo [44].

### 3.3.5.- Carcasa – Impresión 3D

Todos los elementos electrónicos del sistema van integrados en una carcasa modular que se compone por cuatro módulos; tres principales y un pie en forma de peana para mantener el dispositivo erguido cuando no hay una pared cerca del posicionamiento óptimo del sensor.

Toda la carcasa está conformada mediante impresión 3D en plástico PLA de color blanco. A continuación, se detallará cada uno de los módulos:

- **Cobertura:** esta pieza dispondrá de un texto con indicaciones que dependen de la calidad del aire medida por el sensor de CO<sub>2</sub>, con letras que podrán estar impresas en 3D o viniladas. La idea es que los leds se coloquen debajo de esta pieza y se pueda ver el color a través de esta, junto con el texto indicativo.

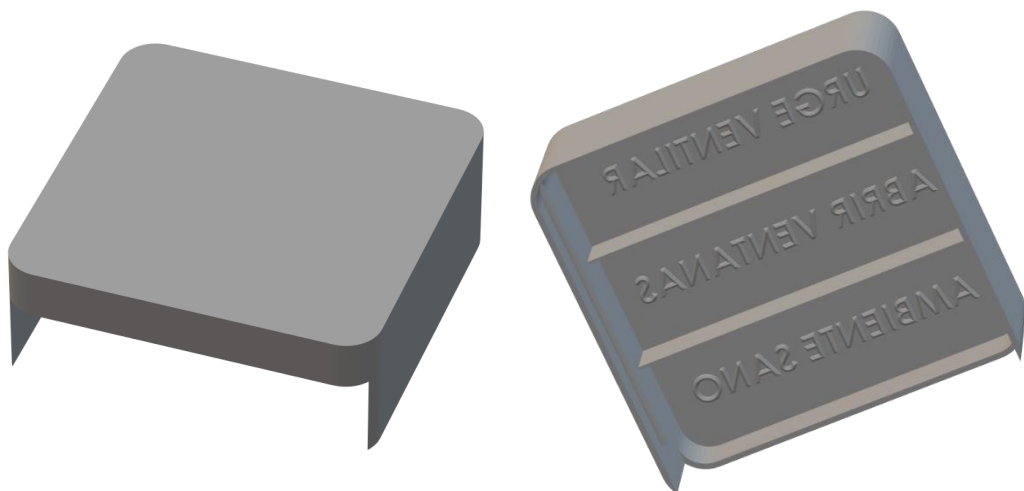


Figura 3.6.- Cobertura.

- **Soporte electrónica:** en esta pieza se atornillará la placa base y se pegarán los leds, cableados desde la placa base, por lo que servirá de base y soporte para toda la electrónica del sistema.

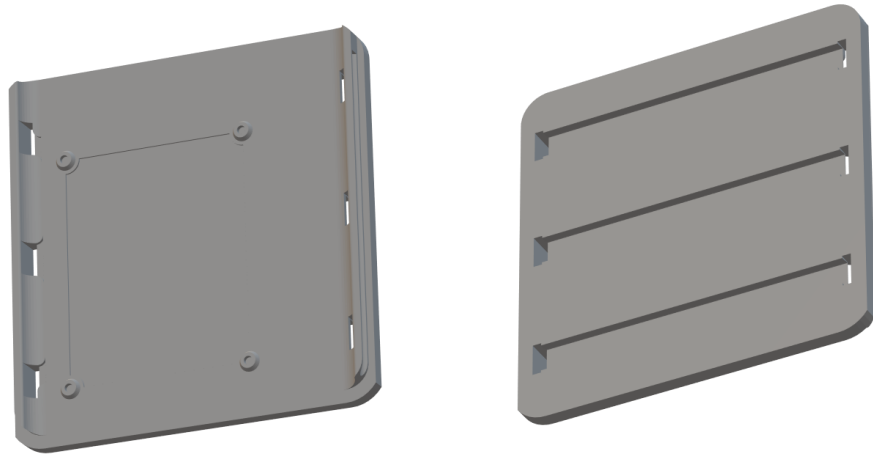


Figura 3.7.- Soporte electrónica.

- **Bastidor:** es la parte de la carcasa que funciona como esqueleto. Dispone de unos orificios para anclarlo a la pared o utilizar una peana para colocar encima de muebles. También cuenta con un hueco para pasar el cable y perforaciones para facilitar el movimiento de aire al interior del medidor.

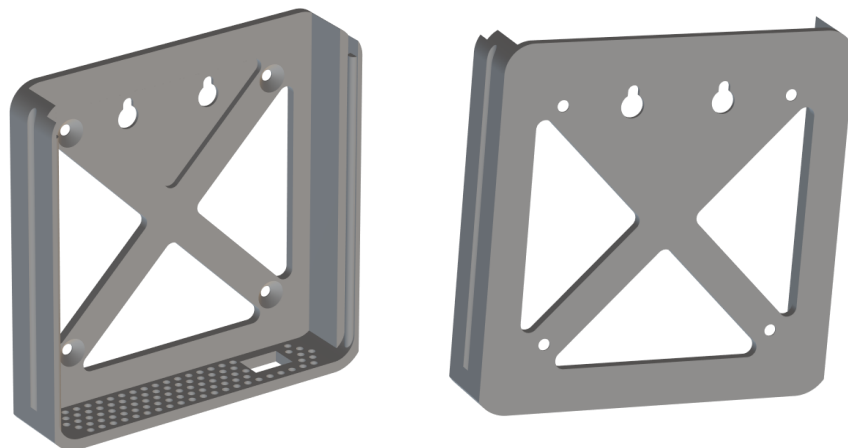


Figura 3.8.- Bastidor.

- **Peana:** esta pieza va a colocada a presión sobre el bastidor y sirve como soporte para que se mantenga de pie el medidor.



Figura 3.9.- Peana.

- **Conjunto bastidor + soporte + difusor:** en la figura 3.9, se muestra el encaje mediante deslizamiento de las tres piezas principales del medidor.



Figura 3.10.- Conjunto bastidor, soporte y difusor.

### 3.3.6.- Placa base – PCB

Todos los elementos electrónicos del sistema irán soldados a una PCB que servirá de comunicación física entre los distintos elementos, creando así un conjunto estable, compacto y estético, a la par que funcional.

En primera instancia, se mostrará el esquema de conexiones que seguirá la placa en la figura 3.10

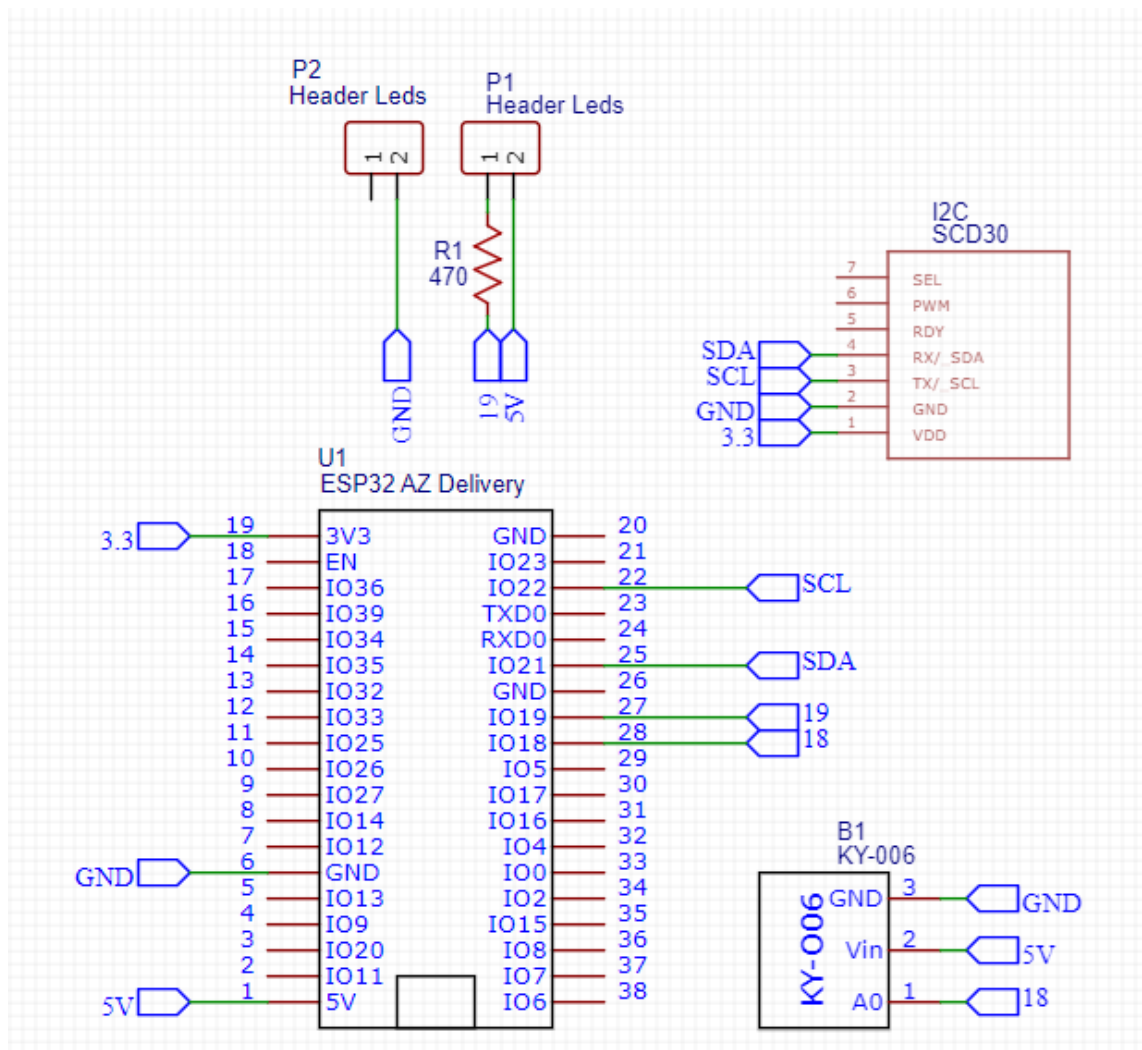


Figura 3.11.- Esquemático de conexiones.

Detallando las conexiones, se puede ver cómo tanto los leds como el zumbador, se alimentan a 5V y se usa el pin 19 (junto con una resistencia) y el pin 18 para el control de los elementos. Por otro lado, se tiene el sensor de SCD30, el cual se estará alimentando desde el pin de 3,3V del ESP32. La conexión de datos se establece a través de los pines de conexión SDA y SCL, con el fin de utilizar el protocolo I<sup>2</sup>C [45].

A partir del esquemático y las conexiones mostradas, se ha diseñado una PCB que, una vez iterado el diseño, ha tenido como resultado la placa mostrada en la figura 3.11. El código QR que se muestra será un enlace que lleva directamente a la aplicación web donde se podrá acceder a los datos y que será descrita más adelante.

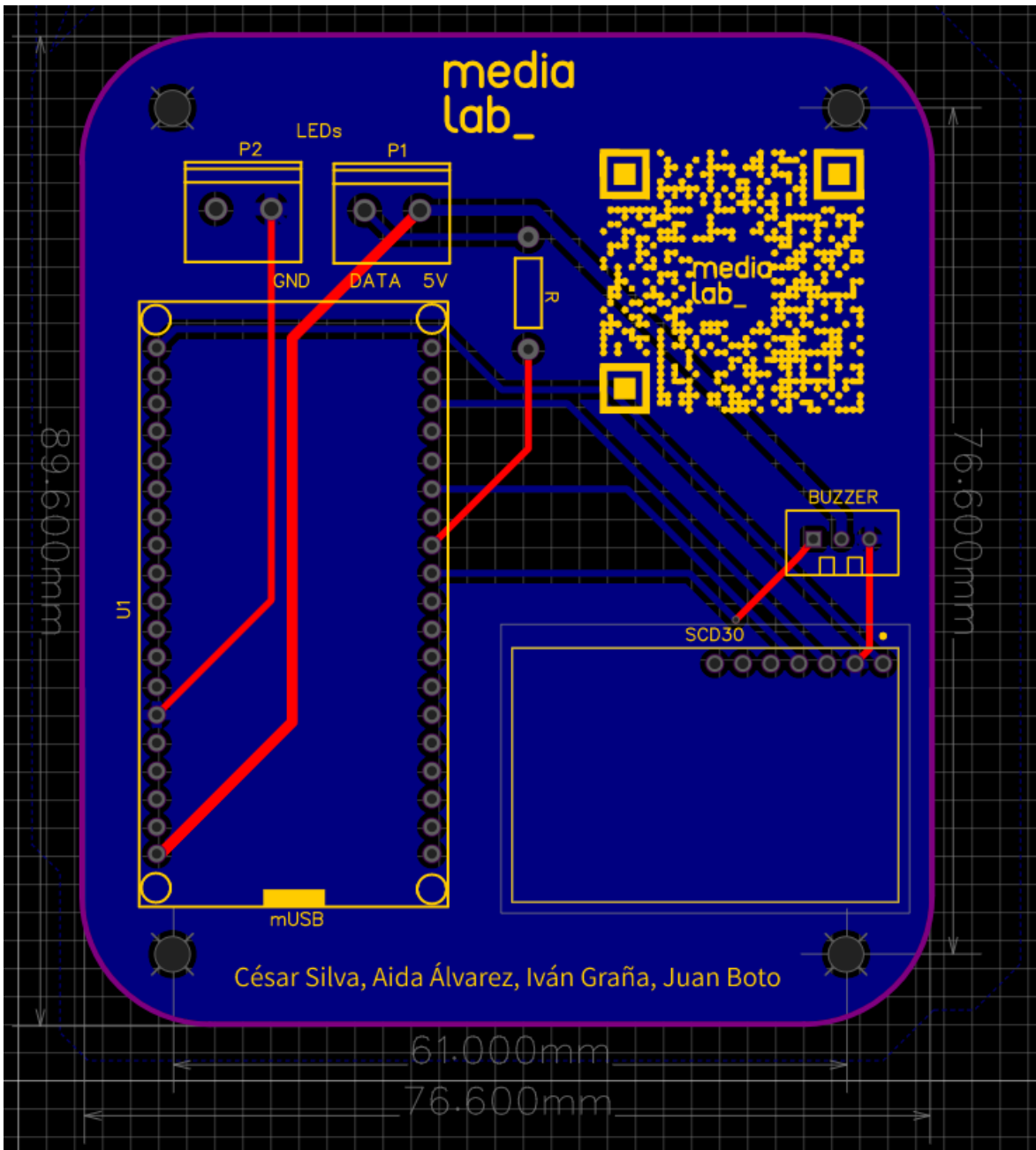


Figura 3.12.- Representación de la PCB en el software EasyEDA.

Con el archivo de fabricación (Gerber) de la PCB, se envía a imprimir a cualquier compañía que se dedique a dicho cometido, como puede ser JCLPCB o PCBWAY, y se obtiene el producto físico.

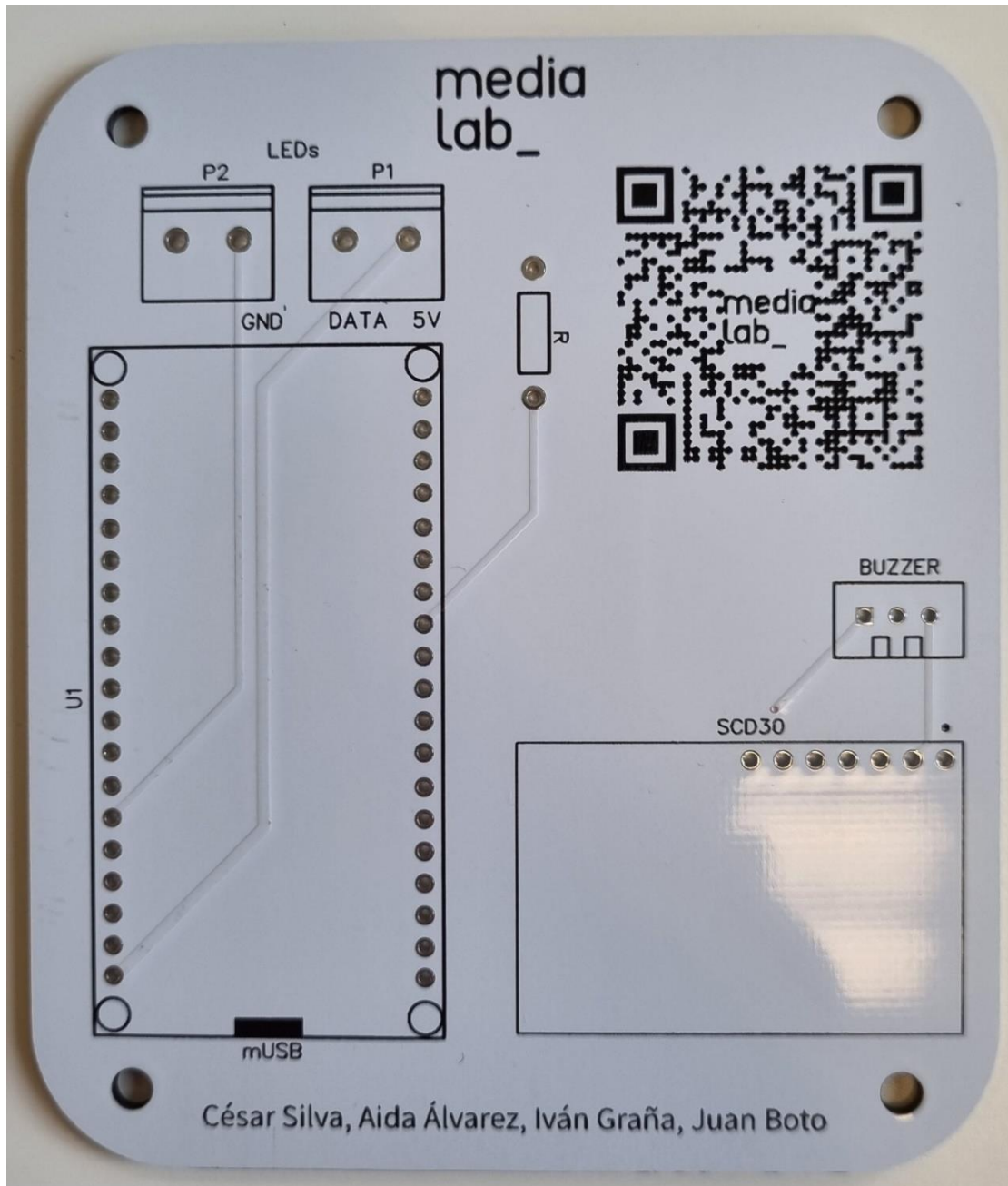


Figura 3.13.- PCB una vez impresa.

Una vez impresa la PCB, se procede a soldar todos los elementos con estaño, obteniendo un conjunto de una sola pieza, utilizando tiras de pines y zócalos hembra para lograr un montaje sencillo y funcional. También se utilizarán diferentes cabezales de PCB para facilitar el cableado de las tiras LED. El resultado puede verse en la placa (figura 3.13).

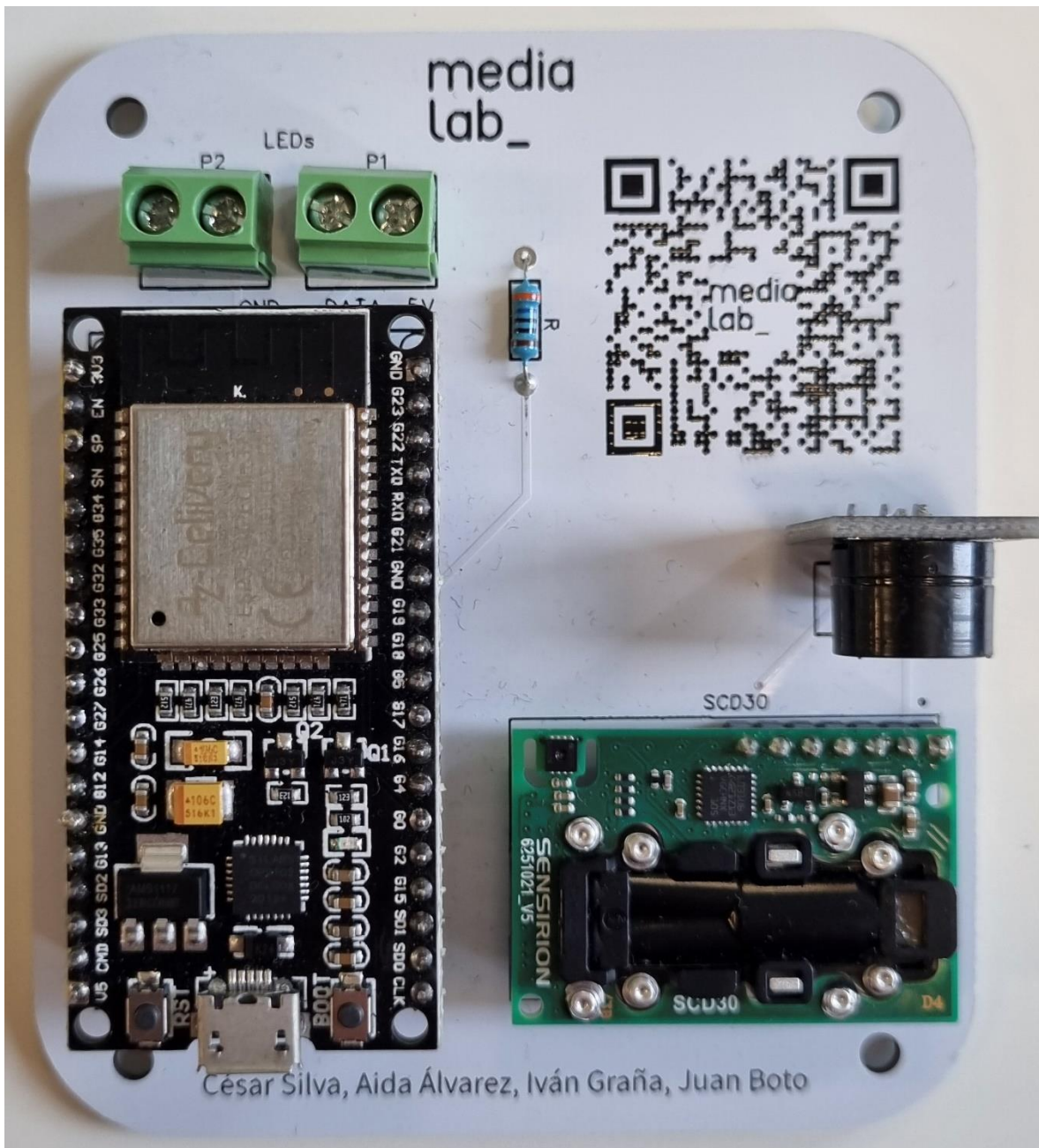


Figura 3.14.- PCB con todos los elementos soldados. Lista para funcionar.



### 3.4.- Funcionamiento

Una vez descritos los elementos del hardware que componen el sensor, se procederá a mostrar cómo está pensado el funcionamiento del sistema. En la figura 3.13 se muestra cómo queda la PCB una vez soldados todos los elementos. Todo estará soldado con la técnica de agujero pasante, lo cual facilita el proceso al hacerlo con un estañador, aprovechando el uso de tiras de pines y zócalos hembra para PCB. De los puertos de leds saldrán los cables que irán conectados a los mismos en serie y la PCB irá atornillada al soporte de electrónica de la carcasa y toda la carcasa permanecerá unida gracias a unos railes por los que se deslizarán las diferentes piezas.

El funcionamiento es muy sencillo de explicar. El sensor de CO<sub>2</sub> obtiene medidas cada dos segundos, las cuales se guardarán hasta obtener 15 medidas y con esas 15 medidas se obtendrá la mediana. El valor de la mediana es el que se utilizará para encender los leds en un color u otro dependiendo de la cantidad del CO<sub>2</sub> en ppm. De esta forma, si la concentración de CO<sub>2</sub> es inferior a 700 ppm, se muestra el verde; si la cantidad de CO<sub>2</sub> se sitúa entre 700 y 1000 ppm, se muestra el ámbar; y si la cantidad de CO<sub>2</sub> es superior a 1000 ppm, se muestra el rojo.

Por otro lado, el pitido funciona de forma que, cuando pasa de un estado de calidad superior a una inferior, se activa la alerta sonora. La alerta sonora se activa una vez de verde a ámbar o dos veces (un pitido doble) si la calidad para de ámbar a rojo o de verde a rojo.

En las siguientes imágenes se muestra como quedaría el conjunto completo cerrado en la carcasa en diferentes situaciones de calidad del aire y como quedarían colocados los leds.



Figura 3.14.- Visualización del sensor completo apagado.



Figura 3.15.- Visualización del sensor mostrando el color verde decretando un ambiente sano.



Figura 3.16.- Visualización del sensor mostrando el color ámbar y sugiriendo abrir ventanas.

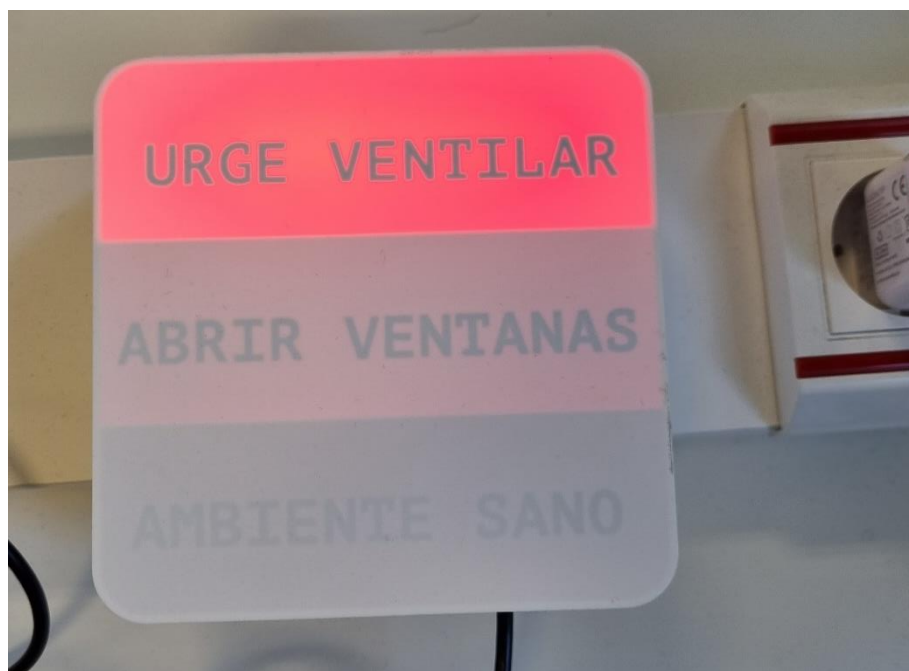


Figura 3.17.- Visualización del sensor mostrando el color rojo y alertando la urgencia de ventilar.



Figura 3.18.- Visualización de la parte posterior del medidor.



Figura 3.19.- Visualización del texto y los LEDs en color ámbar.

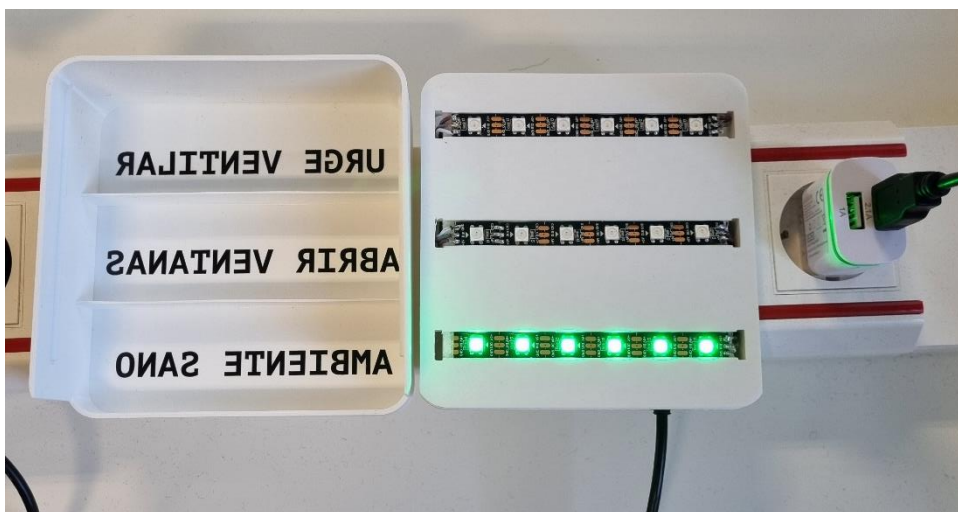


Figura 3.20.- Visualización del texto y los LEDs en color verde.

El funcionamiento de los equipos de medición ha sido comprobado mediante la fabricación de más de quince dispositivos y ha sido monitorizado durante seis meses.



Figura 3.21.- Conjunto de sensores funcionando de forma continua.

### 3.4.1.- Consumo

Con el objetivo de comprobar cual es el consumo de los equipos de medición de CO<sub>2</sub> de los que trata este proyecto, se ha utilizado un comprobador de potencia capaz de registrar la demanda energética a través del adaptador de corriente. Este dispositivo se conecta en serie, entre el propio adaptador y el equipo de medición del cual hacemos uso.

Las condiciones de esta prueba de consumo se han realizado con todas las características del sistema activadas, es decir, procesador a pleno rendimiento, zumbador y leds activados, sensor tomando datos cada dos segundos y la conexión a la red wifi activada y funcionando. También cabe destacar que los leds se han encendido a pleno potencia y no al 40% potencia como se ha diseñado el sistema, para de esta forma ver la energía máxima que consumo el dispositivo si necesitásemos más potencia lumínica para otros ambientes.

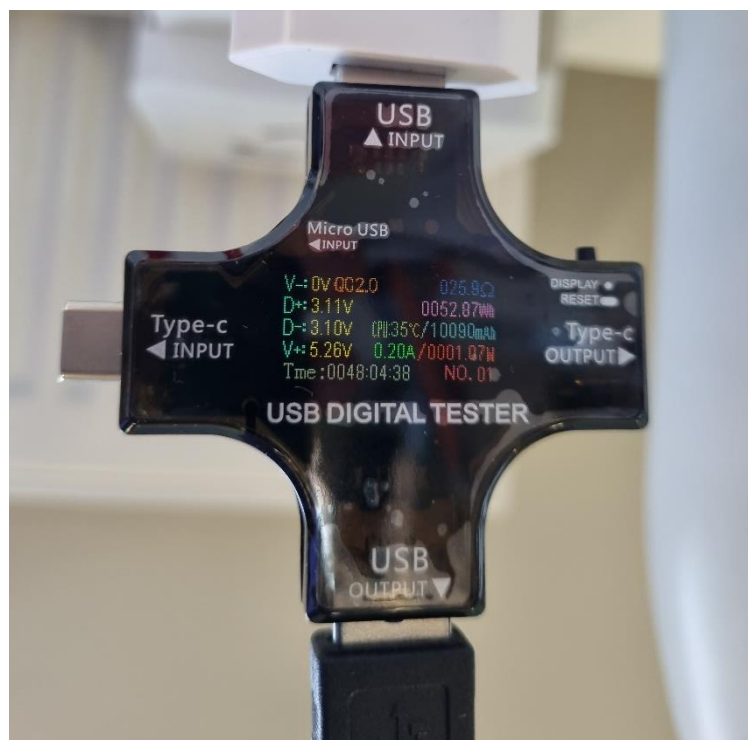


Figura 3.22.- Conjunto de sensores funcionando de forma continua.

El resultado ha sido de 52,47 Wh a las 48 horas y 5 minutos de ensayo, lo que supone un consumo energético de 26,20 Wh al día y de aproximadamente 1,1 Wh a la hora. Sabiendo que el consumo real será inferior (aproximadamente 0,8 Wh a la hora) al no utilizar el máximo brillo permitido por los leds

### 3.5.- Software

En este apartado, quedará descrito el funcionamiento a nivel software del sistema a construir para que funcione de la forma descrita en el apartado 3.4. Ello implica la programación del microprocesador, el envío de los datos a una base de datos y la visualización de dichos datos en una aplicación web.

#### 3.5.1.- Programación ESP32

Tal y como se explicó con anterioridad en el apartado 3.3.4, una de las bondades del ESP32 es el ser compatible con la programación en Arduino, lo que facilita mucho este cometido debido a la amplia disponibilidad de librerías y diferentes atajos que permiten trabajar con los elementos hardware seleccionados de forma completa y flexible. Aprovechando dicha compatibilidad, se utilizará el Arduino IDE 1.8.16.

Es necesario diferenciar entre dos códigos diferentes que se podrán implementar para el medidor de CO<sub>2</sub>. Se va a diferenciar entre código para el funcionamiento local y código Wifi:

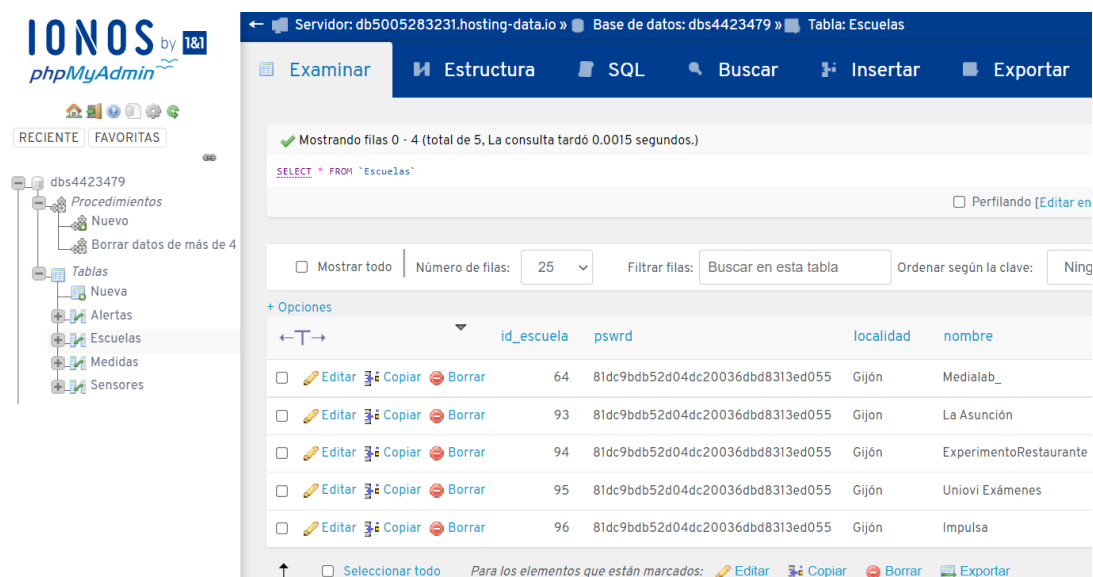
- **Código Local:** este código funciona de forma aislada y dispone de todo lo necesario para tomar las medidas de CO<sub>2</sub> y, en función de este valor, cambiar el color de los leds y emitir alertas sonoras cuando este valor asciende rebasando ciertos umbrales. Lo único necesario para este funcionamiento, es una toma de corriente para alimentar el sistema. Este código estará disponible en el Anexo I.
- **Código Wifi:** este es el código más completo. Requiere de la toma de corriente y una conexión Wifi, pues será a través de este sistema que se enviarán los datos a la base de datos para su posterior gestión y visualización. Dispone de todo lo que disponía el Código Local, además de conexión wifi, otorgándole las credenciales de la red empleada, y envío de datos a la base de datos. Este código estará disponible en el Anexo II y está escrito en dos ficheros diferentes: por un lado, un .ino con las funciones y el bucle principal y por otro lado un archivo de cabecero .h en el que se deberán introducir las credenciales de la red wifi a la que se desea conectar. El funcionamiento es igual que en la versión “local”, a

excepción de que aquí los valores de CO<sub>2</sub> se guardan en una tabla y cuando se llega a 15 valores, se realiza la mediana y es la que se sube a la base de datos. Los siguientes apartados 3.5.2 Base de datos y 3.5.3 Aplicación web, solo tienen sentido si se utiliza este código.

### 3.5.2.- Base de datos

En el caso de decidir que se quiere obtener la funcionalidad completa del sensor y conectarlo a una red wifi, se puede disfrutar de una característica de gran interés que consiste en poder visualizar todos los datos en forma de gráficas históricas y analizar los resultados que se van obteniendo. De esta forma se puede actuar sobre los mecanismos de ventilación de una sala cuando se sepa que se van a exceder los umbrales de calidad de aire bueno, siendo este uno de los objetivos finales de este sistema.

El primer paso para la visualización de datos es recopilar esos datos de forma ordenada. Para ello, se utilizará la base de datos MySQL del servicio de hosting IONOS, servicio que dispone la cátedra MediaLab y que se utilizará para dicho cometido. Se trabajará con la aplicación phpMyAdmin, que está disponible en el citado IONOS para administrar la base de datos[46].



The screenshot shows the phpMyAdmin interface for a MySQL database. The left sidebar shows the database structure with a tree view containing 'Procedimientos' (Nuevo, Borrar datos de más de 4) and 'Tablas' (Nueva, Alertas, Escuelas, Medidas, Sensores). The main area displays the 'Escuelas' table with the following data:

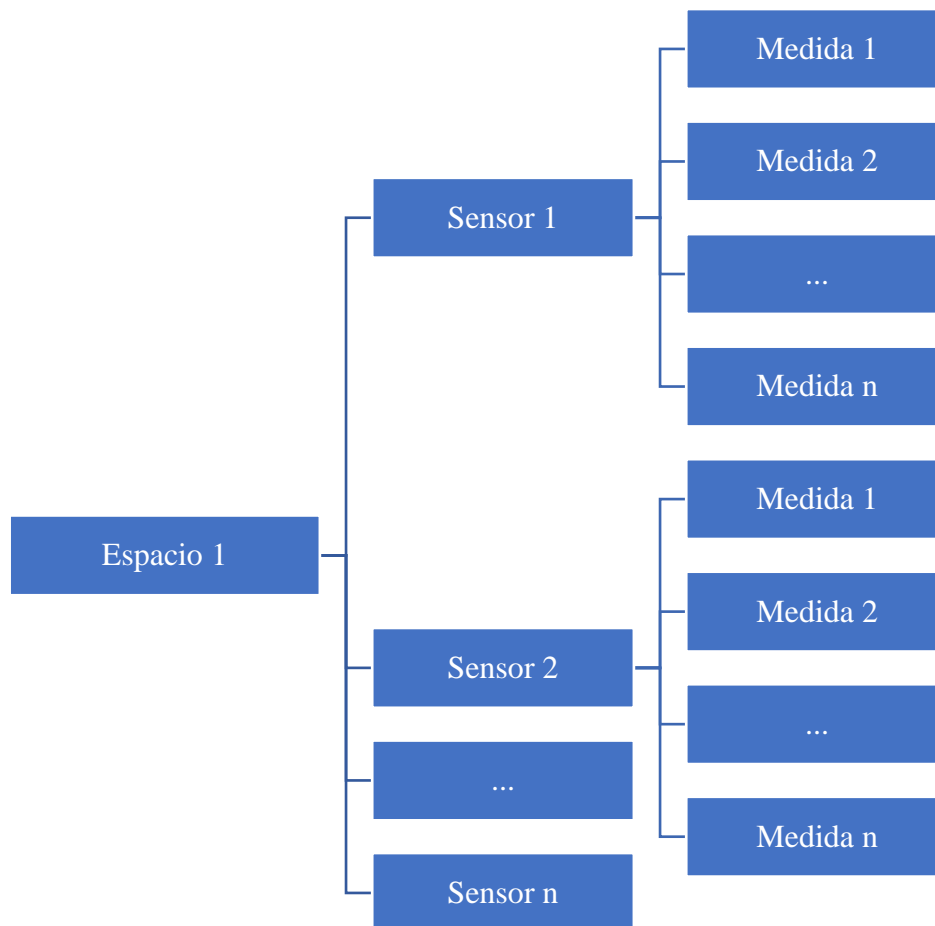
	id_escuela	pswr	localidad	nombre
<input type="checkbox"/>	64	81dc9bdb52d04dc20036dbd8313ed055	Gijón	Medialab_
<input type="checkbox"/>	93	81dc9bdb52d04dc20036dbd8313ed055	Gijón	La Asunción
<input type="checkbox"/>	94	81dc9bdb52d04dc20036dbd8313ed055	Gijón	ExperimentoRestaurante
<input type="checkbox"/>	95	81dc9bdb52d04dc20036dbd8313ed055	Gijón	Uniovi Exámenes
<input type="checkbox"/>	96	81dc9bdb52d04dc20036dbd8313ed055	Gijón	Impulsa

Figura 3.23.- Base de datos en IONOS.



Dentro de la base de datos, existen dos secciones. Una de ellas dedicada a **procedimientos**, donde utilizando pequeños scripts se podrán realizar acciones de limpieza o de cualquier índole dentro de la base de datos. La segunda sección, la principal, será la de las **tablas**, las cuales estarán organizadas de forma jerárquica; esta jerarquía hace referencia a una organización de 1 a “n”; se tiene un número finito de espacios donde se colocarán los sensores y se pueden añadir más espacios en caso de ser necesario. Un espacio, tiene “n” sensores y un sensor tiene “n” medidas. Además, se dispondrá de la “tabla de alertas” donde se guardarán ciertas alertas para futuras indicaciones sobre la calidad del aire.

En el esquema 3.2 se puede ver la jerarquía que tiene un espacio de forma más visual. Hay que recordar que se pueden tener multitud de espacios disponibles. El nombre “escuela” hace referencia a espacios. Se llama “escuela” porque en un principio el proyecto iba dirigido únicamente a centros educativos.



Esquema 3.2.- Jerarquía del espacio.

### 3.5.3.- Aplicación Web

Como se ha mencionado anteriormente, con el objetivo de visualizar los datos recogidos por los sensores en la base de datos, se utilizará una aplicación web que acceda a dichos datos y los muestre de forma ordenada para cada uno de los diferentes sitios o espacios donde se encuentren situados los sensores.

Cada espacio donde se encuentren los sensores dispondrá de una cuenta en la aplicación con sus credenciales propias para poder acceder a todos los sensores que se encuentren disponibles y a cada una de sus medidas en un periodo de tiempo determinado.

Dicha aplicación web estará alojada en la página web de la cátedra MediaLab [47], en una sección dedicada al sensor de CO<sub>2</sub>, en la cual se puede visualizar la siguiente interfaz para acceder a la aplicación.

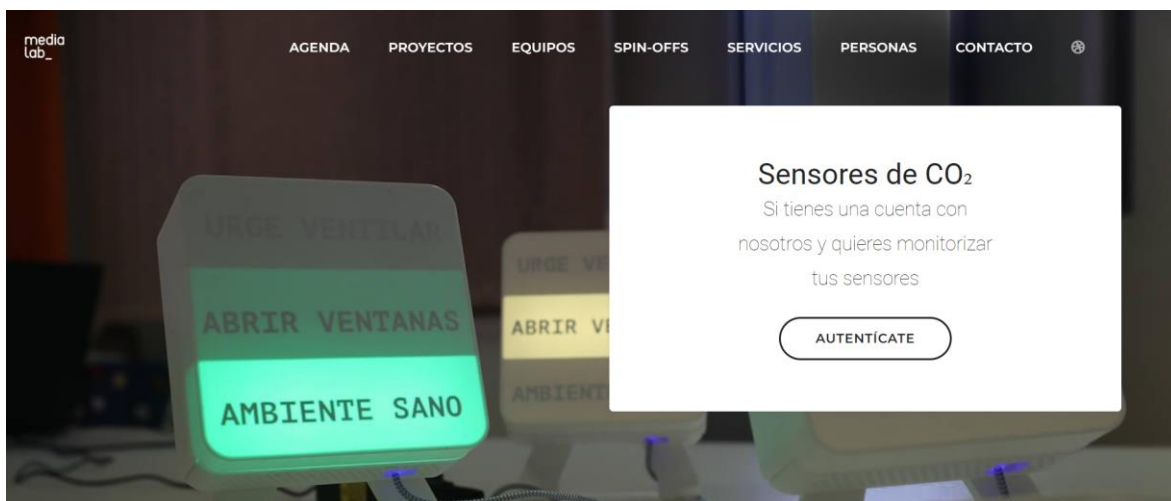


Figura 3.24.- Aplicación web sensor CO<sub>2</sub> Medialab.

Accediendo con las credenciales de prueba de una cuenta denominada MediaLab\_, se pueden visualizar dos sensores que están funcionando desde hace un tiempo a modo de prueba en el espacio de la cátedra MediaLab.

Lo primero que se indica es una lista de sensores como la que se ve en la figura 3.25.

🏠 Medialab\_

🔄 Actualizar tabla

Lista de sensores:

Nombre	Estancia	Última medida CO <sub>2</sub>	Nivel	Temperatura	Humedad	Fecha última medida	Funcionamiento
<a href="#">Sensor_02</a>	Mesas	706 (P.P.M.)	Precaución	23.7 °C	41 %	09 mar, 11:56:23	Correcto
<a href="#">Sensor_06</a>	Estantería	595 (P.P.M.)	Aceptable	24.2 °C	40 %	09 mar, 11:56:26	Correcto

Figura 3.25.- Lista de sensores.

Si se entra en cada uno de los sensores, se puede observar un resumen con todos los datos del sensor, así como el porcentaje de tiempo que pasa en cada uno de los estados, como puede verse en la figura 3.26.

🏠 Medialab\_ / [Sensor\\_06 \(Estantería\)](#) / Resumen

### Estado del sensor durante la última semana

Nombre: **Sensor\_06**

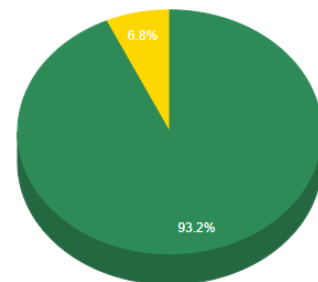
Estancia: **Estantería** [\(Modifica el emplazamiento\)](#)

MAC del sensor: **08:3A:F2:6E:2A:40**

Máximo valor: **901 p.p.m.**

Fecha máximo valor: **2022-03-09 10:46:26**

Tiempo en cada nivel entre semana en horario laboral (de 8 a 20 horas)



● Aceptable (menos de 700 p.p.m.) ● Precaución (entre 700 y 1000 p.p.m.)

Figura 3.26.- Resumen estado sensor.

Como se puede observar en la figura 3.27, también permite elegir el tiempo histórico de muestra de datos y cambiar entre diferentes datos de visualización; concentración de CO<sub>2</sub>, temperatura y humedad. De esta forma, es posible obtener gráficas muy detalladas de la evolución de los datos obtenidos por un sensor en concreto.

[Medialab](#) / [Sensor\\_06\(Estantería\)](#) / Gráficas

Seleccione la medida:

CO<sub>2</sub>  Temperatura  Humedad

Seleccione el tiempo:

8 horas  24 horas  2 semanas  4 meses



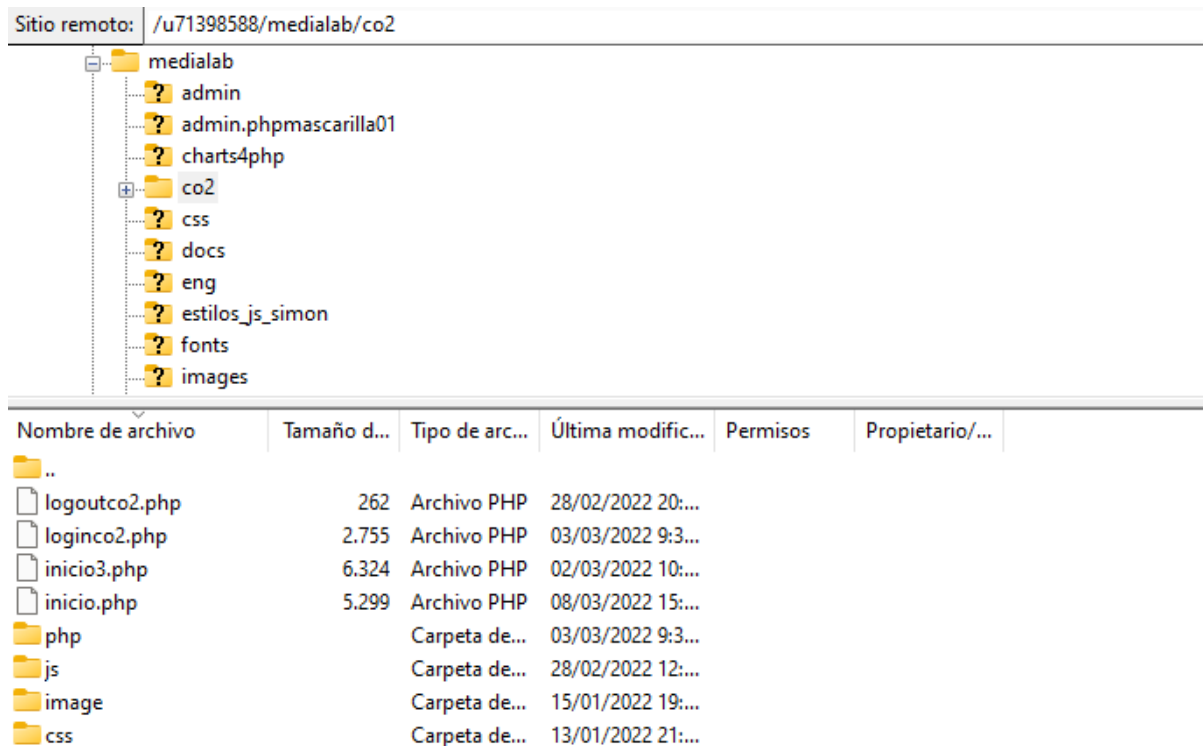
Figura 3.27.- Gráficos de los sensores según diferentes medidas y tiempos.

Una vez mostrado lo que ve el usuario en el front-end (programado en JavaScript, HTML y PHP), resulta interesante describir como está conformado en back-end, puesto que es sumamente importante para el funcionamiento del programa.

El back-end está programado en PHP, lo cual resulta útil para realizar la conexión de la página web con la base de datos. Para gestionar el back-end se utilizará la herramienta FTP filezilla [48], gratuita y sencilla para estos cometidos.

Una vez conectados al servidor FTP, es posible acceder a todos los archivos de programación del servidor, pudiendo modificarlos a nuestro antojo y cambiando cualquier detalle de la página web.

Sitio remoto: /u71398588/medialab/co2



Nombre de archivo	Tamaño d...	Tipo de arc...	Última modific...	Permisos	Propietario/...
..					
logoutco2.php	262	Archivo PHP	28/02/2022 20:...		
loginco2.php	2.755	Archivo PHP	03/03/2022 9:3...		
inicio3.php	6.324	Archivo PHP	02/03/2022 10:...		
inicio.php	5.299	Archivo PHP	08/03/2022 15:...		
php		Carpeta de...	03/03/2022 9:3...		
js		Carpeta de...	28/02/2022 12:...		
image		Carpeta de...	15/01/2022 19:...		
css		Carpeta de...	13/01/2022 21:...		

Figura 3.28.- Servidor FTP.

De interés resulta el archivo loginco2.php, donde se comprueban las credenciales. Al comprobarse, se accede a inicio.php que a su vez llama a inicio.js, que es el archivo JavaScript que contiene todo el contenido de las tablas y contenidos de la aplicación web.

## 4. Resultados experimentales

En esta sección se mostrarán los resultados obtenidos mediante experimentos sobre el funcionamiento de los medidores de CO<sub>2</sub>. Se tratará de comprobar su funcionamiento y constatar que realmente, a más personas en un espacio, suben los niveles de CO<sub>2</sub> y, por otro lado, se tratará de arrojar luz sobre algunos asuntos como la cantidad de sensores necesarios para un espacio o la altura a la que deben colocarse los sensores para obtener medidas más eficaces.

Para obtener los resultados y comprobar el funcionamiento de los sensores, se han llevado a cabo dos experimentos dentro de la Escuela Politécnica de Ingeniería de Gijón. El primero de ellos en el espacio de la Cátedra MediaLab y el segundo en el Aula de Exámenes 2 del Edificio Polivalente.

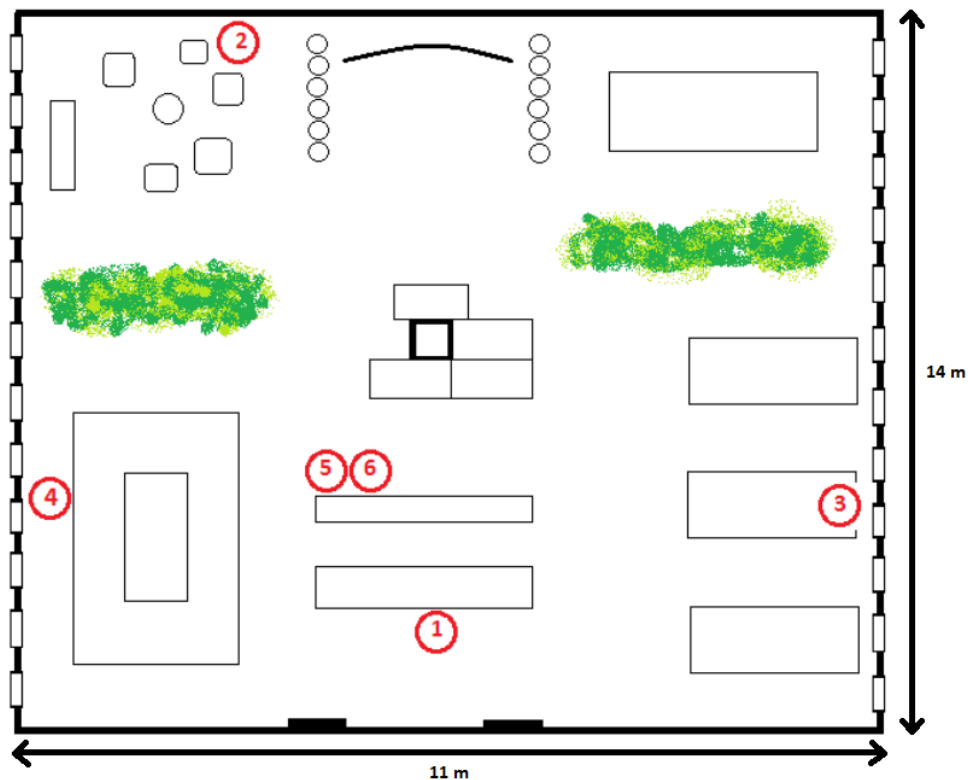
Con el fin de realizar el experimento, se han construido varios medidores siguiendo las pautas establecidas en la sección 3. A partir de este momento, se considerará cada experimento por separado y se tratarán de poner en conjunto las conclusiones obtenidas de ambos tras analizar los resultados arrojados.

Cabe recordar que los sensores toman valores cada dos segundos y obtienen la mediana por cada 15 medidas consecutivas. La mediana es el valor mostrado en las gráficas y el valor que se tomará como la concentración de CO<sub>2</sub>.

### 4.1.- Experimento en espacio MediaLab

Este experimento ha sido desarrollado entre los días 12 de noviembre al 22 de diciembre de 2021, colocando seis sensores situados en el espacio MediaLab, con una superficie de 155,62 m<sup>2</sup>, ocupando los lugares que se pueden ver en la figura 4.1.

La idea es que durante estos días se puedan obtener datos de la variación del CO<sub>2</sub> cuando haya personas en el local y cuando esté vacío, por lo que se ha monitorizado el número de personas en cada zona de la sala, así como la apertura de ventanas.



Esquema 4.1.- Espacio MediaLab con situación de cada uno de los sensores.

1. Entrada
2. Zona descanso
3. Zona trabajo
4. Zona proyector
5. Centro sala (altura 1,5 m)
6. Centro sala (altura 2 m)

Analizando la gráfica 1 del Anexo III, se ven reflejados en un mismo gráfico los datos de todos los sensores de la estancia. Se advierten picos en los valores de CO<sub>2</sub> los días correspondientes con una actividad laboral. Esto sería del 23/11 al 26/11, del 29/11 al 03/12, el 07/12, los días 09/12 y 10/12, del 13/12 al 17/12 y del 20/12 al 22/12. Estos días los niveles de CO<sub>2</sub> varían entre los 400 ppm y los 1200 ppm, tomando valores más realistas y estables a medida que avanza el experimento, debido a que la primera semana se atraviesa una fase de calibración y estabilización. Al final de éste, los valores en periodos de trabajo fluctúan entre los 400 ppm y los 900 ppm. Cuando los sensores superaban los 700 ppm, se abrían las ventanas y de esta forma se mantenían valores por debajo de los 1000 ppm pese a haber una alta concentración de personas en determinadas áreas del MediaLab.

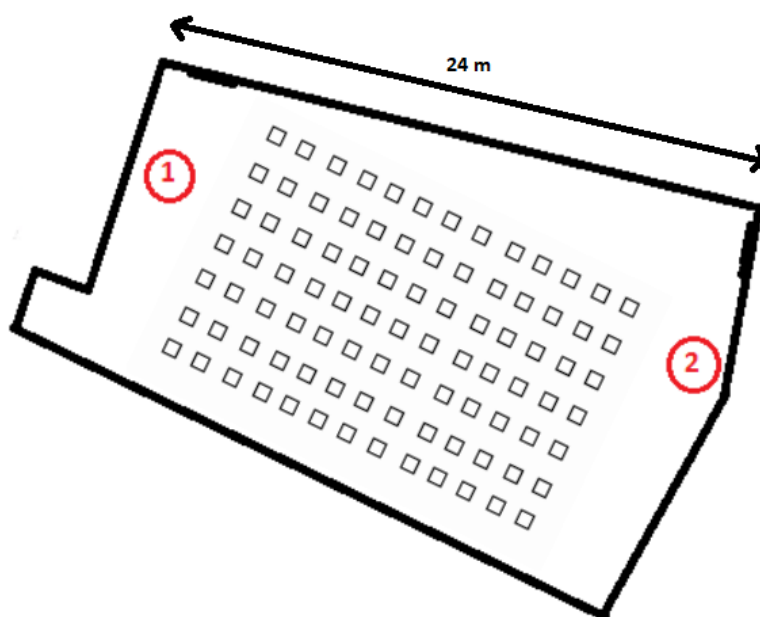
Entre estas etapas los niveles de CO<sub>2</sub> bajan a valores de 400 ppm - 500 ppm, coincidiendo con fines de semana, y en el caso de la primera semana de diciembre, con el puente de la Constitución. Aquí los sensores se estabilizan y sus medidas se toman más lineales.

También se observa una fase irregular en el inicio de la toma de datos, durante los primeros diez días los valores obtenidos no han sido fiables puesto que el sensor se encontraba en una fase de calibración. Además, durante esta etapa los sensores estaban montados sobre protoboards, lo que lo hacía que tomaran medidas más inestables por los cables. Tras los 10 primeros días, se han soldado sobre PCB, lo que ha arreglado cualquier mala conexión.

Después de esta fase de calibración del 12/11 al 22/11 seguido de dos días sin datos debido a un corte de luz en la estancia, también se aprecia claramente cuando un sensor ha sufrido alguna interrupción y se ha desconectado, dejando de obtener datos.

Respecto del número de sensores, se puede apreciar una correlación entre todos, por lo que tener seis sensores dentro de una estancia con la superficie como la del MediaLab carece de sentido. Situar uno en el centro, o dos, uno en cada esquina, alejados de corrientes de aire, resultaría suficiente.

#### 4.2.- Experimento Aula Exámenes



Esquema 4.2.- Aula de exámenes con situación de los dos sensores.



1. Entrada
2. Zona ventanas

Este experimento ha sido desarrollado entre los días 10 de enero al 22 de enero de 2022, colocando dos sensores situados en un aula de exámenes del Edificio Polivalente de la Escuela Politécnica de Ingeniería de Gijón, de forma que los cada uno de los sensores se queda colocado en cada extremo del aula, uno en la entrada y otro debajo de la zona de ventanas. El aula tiene una superficie total de 489,34 m<sup>2</sup> y su forma puede verse en la figura 4.2.

Este experimento se ha llevado a cabo con la idea de comprobar cómo afecta o no el hecho de que haya personas entrando en un aula de gran tamaño y comprobar si los niveles de CO<sub>2</sub> suben con la presencia de personas, comprobando una de las hipótesis iniciales.

Durante el periodo en el cual se ha llevado a cabo el experimento, se han realizado en el aula siete exámenes:

- 13/01 por la tarde.
- 14/01 por la mañana.
- 14/01 por la tarde.
- 17/01 por la mañana.
- 18/01 por la mañana.
- 19/01 por la mañana.
- 19/01 por la tarde.

Observando la gráfica 2 del Anexo III, se pueden analizar los resultados durante estos días. En primer lugar, remarcar que, entre un examen y otro, las ventanas y la puerta permanecen abiertas, por lo que entra en juego la ventilación y es posible obtener valores de CO<sub>2</sub> similares a los del aire de la atmosfera. Se puede observar que cuando no hay nadie en el aula, los valores se sitúan entre 400 ppm y 500 ppm. Sabiendo que, como se mencionaba en la sección 1, los valores de CO<sub>2</sub> promedio en la atmósfera rondan los 417 ppm, justificando que los valores con el aula vacía concuerdan con los valores esperados.

En las franjas en las que ha habido gente en el aula, durante los periodos correspondientes a la realización de exámenes, puede observarse como los valores de CO<sub>2</sub> suben y llega a haber

concentraciones que rondan los 700 ppm, lo cual no es un valor muy alto, pero a partir del cual ya conviene llevar protocolos de ventilación más agresivos. La puerta permanece abierta durante los exámenes, al igual que alguna ventana, lo que junto con el gran volumen del aula ayuda a no obtener concentraciones mayores de CO<sub>2</sub> y disfrutar de un ambiente correcto.

Durante los días 15 y 16, correspondientes al fin de semana, se observa como los valores se estabilizan y las medidas se igualan.

Respecto del número de sensores, se puede ver que los dos sensores suben de manera correlacionada, aunque es cierto que en ocasiones un sensor obtiene valores mayores que el otro, lo que puede ser debido a mayor concentración de personas más cercanas al área, a la incidencia directa del sol o a diferentes factores como corrientes de aire que pueden afectar a la medición. Se observa que con un solo sensor se tendría ya una referencia para abrir ventanas.

### **4.3.- Resultados**

Una vez estudiados los resultados, se comprueba que los sensores cumplen su cometido de monitorizar el CO<sub>2</sub>, comprobando que cuando hay personas, la cantidad de CO<sub>2</sub> sube como es esperado, y ayudan a tomar decisiones sobre la ventilación, teniendo unas medidas eficaces que facilitan dicha toma de decisiones.

También cabe resaltar que el área de influencia de un sensor es más grande del esperado. Si bien para un área de gran tamaño conviene tener más de un sensor, se puede ver como las concentraciones de CO<sub>2</sub> suben de manera correlacionada.

Respecto de la altura de colocación, se puede comparar el sensor 5 y el sensor 6 del experimento de MediaLab, comprobando que la altura no tiene un impacto significativo en el rendimiento del sensor. Para este cometido, se han colocado a la misma altura durante una semana y luego se han separado medio metro en el plano de altura.

# 5. Conclusiones y futuro

## 5.1.- Conclusiones

En primer lugar, mencionar el alto grado de satisfacción con los resultados obtenidos al haber completado los objetivos iniciales que se planteaban para la resolución de este proyecto.

Del trabajo realizado se extraen conclusiones muy interesantes. En primera instancia, cabe resaltar que, como se ha demostrado, es posible fabricar un producto con unas características muy buenas, mejores que las de medidores de su mismo rango de precio, y a la vez con un sistema completamente transparente. La mayoría de los medidores del mercado funcionan como una “caja negra” en el sentido de que no hacen ningún tipo de alusión al sensor que utilizan y, por ende, se desconocen sus características reales.

En la misma línea, tampoco se dispone por norma general de una manera de recalibrar los sensores de los medidores comerciales de forma rápida. Se sabe que los movimientos y golpes mecánicos, entre otras cosas, pueden afectar a la calibración de los sensores NDIR [49]. Por ese motivo, disponer de un sistema de calibración automática o una manera de poder recalibrar los sensores de forma rápida, resulta ciertamente interesante para asegurar que las medidas son lo más fieles posible a la realidad. El sistema diseñado la tiene y tiene la opción de recalibración manual de forma sencilla.

También resulta interesante confirmar que cumplen el propósito para el que se ha fabricado y ayuda a tener consciencia a la hora de actuar sobre los sistemas de ventilación, así como conocer la calidad de aire de una estancia. En esencia, se ha verificado que, a más personas, más CO<sub>2</sub> se puede medir en el ambiente lo que permite tomar acciones sobre los elementos de ventilación. Si bien no resulta tan interesante tomar el valor de la concentración de CO<sub>2</sub> en ppm como una verdad absoluta, pues se sabe que, como todo sensor, está sometido a diversos tipos de error y variables que afectan tanto a la medida como a la calibración (temperatura, corrientes de aire, movimientos bruscos, etc.); por lo que resulta extremadamente difícil de controlar todo ese ambiente que afecta a la medida correcta del CO<sub>2</sub>, se puede tomar el valor como una estimación precisa y comprobar como realmente, mirando los resultados de

forma global, se comprueban las hipótesis que se planteaban a comienzos de redacción de este documento.

Como se ha comprobado dentro del experimento realizado en el MediaLab, el conocer los valores de CO<sub>2</sub> en el espacio, ha servido de forma efectiva para tener una referencia sobre cuando abrir las ventanas para mantener unos niveles de CO<sub>2</sub> correctos. Teniendo en cuenta que el experimento se ha llevado a cabo durante un periodo frío del año, no ha sido necesario tener las ventanas abiertas durante todo el tiempo, favoreciendo la comodidad de la gente que trabaja en el espacio sin renunciar a tener un espacio seguro.

Dicho esto, se puede concluir que el uso de sensores de CO<sub>2</sub> favorece la correcta ventilación de las aulas o espacios en los que son colocados y, además, remarcar el interés que tienen como forma de controlar la expansión de patógenos que se transmiten por vía aérea. Su uso puede facilitar las cosas a partir de ahora, de cara a nuevas pandemias o enfermedades estacionarias, ayudando a tener ambientes más sanos, sabiendo que un exceso de CO<sub>2</sub> en un espacio cerrado conlleva una serie de problemas para la salud que están presentes, aunque no haya patógenos en el aire [50].

## 5.2.- Trabajos futuros

Habiendo creado un sensor completo con unas características de lo más competitivas respecto de la situación actual del mercado, las puertas abiertas para trabajos futuros no son precisamente amplias, pero siempre hay lugar a la mejora y al progreso continuo, por lo que se mencionarán varios campos en los que puede continuar enfocándose el desarrollo:

- Inclusión de nuevos protocolos de comunicación para sustituir o complementar la conexión Wifi, como pueden ser LoraWAN [51] o 6LoWPAN [52].
- Inclusión de nuevos sensores para valorar la calidad del aire y no solo depender de un sensor de CO<sub>2</sub>.
- Implementar y desarrollar un software para facilitar la instalación del firmware en los medidores sin necesidad de tocar código.
- Implementar un protocolo seguro de actualizaciones vía OTA [53].

## 6. Referencias y bibliografía

- [1] M. Ehsanifar, "Airborne aerosols particles and COVID-19 transition," *Environmental Research*, vol. 200, p. 111752, Sep. 2021, doi: 10.1016/J.ENVRES.2021.111752.
- [2] Z. Peng and J. L. Jimenez, "Exhaled CO<sub>2</sub> as a COVID-19 infection risk proxy for different indoor environments and activities," *Environmental Science and Technology Letters*, vol. 8, no. 5, pp. 392–397, May 2021, doi: 10.1021/ACS.ESTLETT.1C00183/SUPPL\_FILE/EZ1C00183\_SI\_001.PDF.
- [3] A. di Gilio *et al.*, "CO<sub>2</sub> concentration monitoring inside educational buildings as a strategic tool to reduce the risk of Sars-CoV-2 airborne transmission," *Environmental Research*, vol. 202, p. 111560, Nov. 2021, doi: 10.1016/J.ENVRES.2021.111560.
- [4] "Díaz Ayuso plantea la obligatoriedad de medidores de CO<sub>2</sub> en la hostelería | Comunidad de Madrid." <https://www.comunidad.madrid/noticias/2020/11/10/diaz-ayuso-plantea-obligatoriedad-medidores-co2-hosteleria> (accessed Dec. 03, 2021).
- [5] "Carbon Dioxide | Vital Signs – Climate Change: Vital Signs of the Planet." <https://climate.nasa.gov/vital-signs/carbon-dioxide/> (accessed Dec. 13, 2021).
- [6] Z. Peng and J. L. Jimenez, "Exhaled CO<sub>2</sub> as a COVID-19 infection risk proxy for different indoor environments and activities," *Environmental Science and Technology Letters*, vol. 8, no. 5, pp. 392–397, May 2021, doi: 10.1021/ACS.ESTLETT.1C00183/SUPPL\_FILE/EZ1C00183\_SI\_001.PDF.
- [7] Z. Liu and J. C. Little, "Materials responsible for formaldehyde and volatile organic compound (VOC) emissions," *Toxicity of Building Materials*, pp. 76–121, Jan. 2012, doi: 10.1533/9780857096357.76.
- [8] ams AG, "ams Datasheet CCS811 Ultra-Low Power Digital Gas Sensor for Monitoring Indoor Air Quality".
- [9] "Operating principle -Catalytic-type gas sensor." <https://www.figaro.co.jp/en/technicalinfo/principle/catalytic-type.html> (accessed Mar. 21, 2022).
- [10] "Wheatstone Bridge Circuit | Theory, Example and Applications." <https://www.electronicshub.org/wheatstone-bridge/> (accessed Mar. 21, 2022).
- [11] M. Struzik *et al.*, "Communication 1804098 (1 of 10) A Simple and Fast Electrochemical CO<sub>2</sub> Sensor Based on Li<sub>7</sub>La<sub>3</sub>Zr<sub>2</sub>O<sub>12</sub> for Environmental Monitoring," 2018, doi: 10.1002/adma.201804098.
- [12] "Ventilación - Entender Covid." <http://www.entendercovid.es/ventilacion/> (accessed Feb. 10, 2022).
- [13] "Schematic of a typical NDIR sensor. The sensor consists of an infrared... | Download Scientific Diagram." [https://www.researchgate.net/figure/Schematic-of-a-typical-NDIR-sensor-The-sensor-consists-of-an-infrared-broadband-source\\_fig1\\_336173057](https://www.researchgate.net/figure/Schematic-of-a-typical-NDIR-sensor-The-sensor-consists-of-an-infrared-broadband-source_fig1_336173057) (accessed Mar. 22, 2022).
- [14] L. B. Mendes, N. W. M. Ogink, N. Edouard, H. J. C. van Dooren, I. de F. F. Tinôco, and J. Mosquera, "NDIR Gas Sensor for Spatial Monitoring of Carbon Dioxide Concentrations in Naturally Ventilated Livestock Buildings," *Sensors 2015, Vol. 15, Pages 11239-11257*, vol. 15, no. 5, pp. 11239–11257, May 2015, doi: 10.3390/S150511239.
- [15] S. Moumen, I. Raible, A. Krauß, and J. Wöllenstein, "Infrared investigation of CO<sub>2</sub> sorption by amine based materials for the development of a NDIR CO<sub>2</sub> sensor," *Sensors and Actuators B: Chemical*, vol. 236, pp. 1083–1090, Nov. 2016, doi: 10.1016/J.SNB.2016.06.014.
- [16] "CO<sub>2</sub> Sensor SCD4x | Sensirion." <https://www.sensirion.com/en/environmental-sensors/carbon-dioxide-sensors/carbon-dioxide-sensor-scd4x/> (accessed Jan. 14, 2022).
- [17] "Detector de Calidad del Aire, Medidor de CO<sub>2</sub> / HCHO/TVOC/TEMP/HUMI Portátil con Pantalla de 3,5 Pulgadas y Sensor de Temperatura y Humedad para Interior y Exterior : Amazon.es: Industria, empresas

y ciencia.” [https://www.amazon.es/Detector-Port%C3%A1til-Pantalla-Pulgadas-Temperatura/dp/B09JR3SJ5S/ref=sr\\_1\\_20?\\_\\_mk\\_es\\_ES=%C3%85M%C3%85C5%BD%C3%95%C3%91&crd=38JHWHEB2NJSJ&keywords=sensor+co2&qid=1647950531&sprefix=sensor+co%2Caps%2C119&sr=8-20](https://www.amazon.es/Detector-Port%C3%A1til-Pantalla-Pulgadas-Temperatura/dp/B09JR3SJ5S/ref=sr_1_20?__mk_es_ES=%C3%85M%C3%85C5%BD%C3%95%C3%91&crd=38JHWHEB2NJSJ&keywords=sensor+co2&qid=1647950531&sprefix=sensor+co%2Caps%2C119&sr=8-20) (accessed Mar. 22, 2022).

- [18] “TACKLY Medidor de co2 ambiente mini- detector co2 con medidor calidad de aire interior - higrometro digital medidor de humedad y termómetro digital para casa - medidor co2 ndir : Amazon.es: Industria, empresas y ciencia.” [https://www.amazon.es/Medidor-ambiente-detector-medidor-interior/dp/B08PD15JSP/ref=sr\\_1\\_11?\\_\\_mk\\_es\\_ES=%C3%85M%C3%85C5%BD%C3%95%C3%91&crd=38JHWHEB2NJSJ&keywords=sensor+co2&qid=1647950531&sprefix=sensor+co%2Caps%2C119&sr=8-11](https://www.amazon.es/Medidor-ambiente-detector-medidor-interior/dp/B08PD15JSP/ref=sr_1_11?__mk_es_ES=%C3%85M%C3%85C5%BD%C3%95%C3%91&crd=38JHWHEB2NJSJ&keywords=sensor+co2&qid=1647950531&sprefix=sensor+co%2Caps%2C119&sr=8-11) (accessed Mar. 22, 2022).
- [19] “Medidor de CO2 portátil | Medidor Profesional | Dioxcare.” [https://dioxcare.com/es/medidores/1-dioxcare-dx700-7427244823227.html?gclid=Cj0KCQjw5-WRBhCKARIsAAId9Fmp-ha8SVHn\\_GJ6lcABkkKcPLig58aHtQV6lPvJBhp-qEuU4mPrn9AaAuOLEALw\\_wcB](https://dioxcare.com/es/medidores/1-dioxcare-dx700-7427244823227.html?gclid=Cj0KCQjw5-WRBhCKARIsAAId9Fmp-ha8SVHn_GJ6lcABkkKcPLig58aHtQV6lPvJBhp-qEuU4mPrn9AaAuOLEALw_wcB) (accessed Mar. 22, 2022).
- [20] “AirGuardian Mini - Medidor inteligente de CO2, humedad y temperatura.” <https://airguardian.es/products/airguardian-mini-2-0> (accessed Mar. 22, 2022).
- [21] “Leading gas sensing solutions supplier in China Intelligent Infrared CO2 Module (Model: MH-Z19) User’s Manual”, Accessed: Apr. 25, 2022. [Online]. Available: [www.winsensor.com](http://www.winsensor.com)
- [22] “How Coronavirus Spreads | CDC.” <https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-covid-spreads.html> (accessed Mar. 24, 2022).
- [23] S. Petersen, K. L. Jensen, A. L. S. Pedersen, and H. S. Rasmussen, “The effect of increased classroom ventilation rate indicated by reduced CO2 concentration on the performance of schoolwork by children,” *Indoor Air*, vol. 26, no. 3, pp. 366–379, Jun. 2016, doi: 10.1111/INA.12210.
- [24] Y. Shen *et al.*, “Community Outbreak Investigation of SARS-CoV-2 Transmission among Bus Riders in Eastern China,” *JAMA Internal Medicine*, vol. 180, no. 12, pp. 1665–1671, Dec. 2020, doi: 10.1001/JAMAINTERNMED.2020.5225.
- [25] S. L. Miller *et al.*, “Transmission of SARS-CoV-2 by inhalation of respiratory aerosol in the Skagit Valley Chorale superspreading event,” *Indoor Air*, vol. 31, no. 2, pp. 314–323, Mar. 2021, doi: 10.1111/INA.12751.
- [26] P. Guijarro Miragaya, T. Estay Leal, L. Patrón Saade, and R. Tendero Caballero, “The CO2 assessment in a school classroom for an optimal natural ventilation strategy = The CO2 assessment in a school classroom for an optimal natural ventilation strategy,” *Building & Management*, vol. 5, no. 3, p. 29, Oct. 2021, doi: 10.20868/BMA.2021.3.4712.
- [27] C. Bonell *et al.*, “An evidence-based theory of change for reducing SARS-CoV-2 transmission in reopened schools,” *Health and Place*, vol. 64, Jul. 2020, doi: 10.1016/J.HEALTHPLACE.2020.102398.
- [28] P. F. Pereira and N. M. M. Ramos, “The impact of mechanical ventilation operation strategies on indoor CO2 concentration and air exchange rates in residential buildings,” *Indoor and Built Environment*, vol. 30, no. 9, pp. 1516–1530, Nov. 2021, doi: 10.1177/1420326X20960767.
- [29] H. Kitamura, Y. Ishigaki, T. Kuriyama, and T. Moritake, “CO2 concentration visualization for COVID-19 infection prevention in concert halls,” *Environmental and Occupational Health Practice*, vol. 3, no. 1, p. n/a, 2021, doi: 10.1539/EOHP.2021-0010-OA.
- [30] J. FRANQUELO SOLER, E. B. BLÁZQUEZ PARRA, O. de COZAR MACIAS, M. D. MARIN GRANADOS, and F. J. SOTO VILLARAN, “NATURALLY VENTILATED CLASSROOMS: CO2 AND COVID PROTECTION,” *DYNA ENERGIA Y SOSTENIBILIDAD*, vol. 11, no. 1, p. [12]-[12], Feb. 2022, doi: 10.6036/ES10341.

- [31] A. Asif, M. Zeeshan, and M. Jahanzaib, "Indoor temperature, relative humidity and CO2 levels assessment in academic buildings with different heating, ventilation and air-conditioning systems," *Building and Environment*, vol. 133, pp. 83–90, Apr. 2018, doi: 10.1016/J.BUILDENV.2018.01.042.
- [32] L. Schibuola, M. Scarpa, and C. Tambani, "Natural Ventilation Level Assessment in a School Building by CO2 Concentration Measures," *Energy Procedia*, vol. 101, pp. 257–264, Nov. 2016, doi: 10.1016/J.EGYPRO.2016.11.033.
- [33] B. Simanic, B. Nordquist, H. Bagge, and D. Johansson, "Indoor air temperatures, CO2 concentrations and ventilation rates: Long-term measurements in newly built low-energy schools in Sweden," *Journal of Building Engineering*, vol. 25, p. 100827, Sep. 2019, doi: 10.1016/J.JOBE.2019.100827.
- [34] G. Pei, D. Rim, S. Schiavon, and M. Vannucci, "Effect of sensor position on the performance of CO2-based demand controlled ventilation," *Energy and Buildings*, vol. 202, p. 109358, Nov. 2019, doi: 10.1016/J.ENBUILD.2019.109358.
- [35] L. Stabile, G. Buonanno, A. Frattolillo, and M. Dell'Isola, "The effect of the ventilation retrofit in a school on CO2, airborne particles, and energy consumptions," *Building and Environment*, vol. 156, pp. 1–11, Jun. 2019, doi: 10.1016/J.BUILDENV.2019.04.001.
- [36] E. Rivas, J. L. Santiago, F. Martín, and A. Martilli, "Impact of natural ventilation on exposure to SARS-CoV 2 in indoor/semi-indoor terraces using CO2 concentrations as a proxy," *Journal of Building Engineering*, vol. 46, p. 103725, Apr. 2022, doi: 10.1016/J.JOBE.2021.103725.
- [37] "The prevention from infection with COVID-19 of students in auditoriums through carbon dioxide measurements – an evidence from Estonian and Latvian high schools," *Safety and Health at Work*, vol. 13, p. S137, Jan. 2022, doi: 10.1016/J.SHAW.2021.12.1179.
- [38] "Datasheet Sensirion SCD30 Sensor Module", Accessed: Mar. 02, 2022. [Online]. Available: [www.sensirion.com](http://www.sensirion.com)
- [39] "Field calibration for SCD30," 2018, Accessed: Apr. 19, 2022. [Online]. Available: [www.sensirion.com](http://www.sensirion.com)
- [40] "Módulo de Alarma de Buzzer Piezo Pasivo KY-006 – AZ-Delivery." <https://www.az-delivery.de/es/products/buzzer-modul-passiv> (accessed Feb. 23, 2022).
- [41] "Sensor Wiki: KY-006 Passive Piezo-Buzzer Module - The Geek Pub." <https://www.thegeekpub.com/wiki/sensor-wiki-ky-006-passive-piezo-buzzer-module/> (accessed Apr. 24, 2022).
- [42] "WS2812B Intelligent control LED integrated light source".
- [43] "Esp - 32 Development Toolkit C Bei Az - delivery.de – AZ-Delivery." <https://www.az-delivery.de/es/products/esp32-developmentboard?variant=36542176914> (accessed Feb. 23, 2022).
- [44] "ESP32 Series Datasheet Including," 2022, Accessed: Apr. 24, 2022. [Online]. Available: <https://www.espressif.com/en/support/download/documents>.
- [45] "I2C-Bus: What's that?" <https://web.archive.org/web/20150422060439/http://www.i2c-bus.org/i2c-bus> (accessed Mar. 24, 2022).
- [46] "IONOS de 1&1 » E-mail, dominios y páginas web." <https://www.ionos.es/> (accessed Mar. 09, 2022).
- [47] "MediaLab de la Universidad de Oviedo." <https://medialab-uniovi.es/pro-sensorco2.php#> (accessed Mar. 09, 2022).
- [48] "FileZilla - The free FTP solution." <https://filezilla-project.org/> (accessed Mar. 10, 2022).
- [49] "Handling Guideline For the SCD30 CO2, humidity and temperature sensor Mechanical Handling," 2016, Accessed: Apr. 19, 2022. [Online]. Available: [www.sensirion.com](http://www.sensirion.com)

- [50] “Dióxido de carbono CO<sub>2</sub> | Instituto para la Salud Geoambiental.”  
<https://www.saludgeoambiental.org/dioxido-carbono-co2> (accessed Apr. 19, 2022).
- [51] “LoRaWAN y LoRa.” <https://lorawan.es/> (accessed Apr. 17, 2022).
- [52] “What is 6LoWPAN for IoT & M2M » Electronics Notes.” <https://www.electronics-notes.com/articles/connectivity/ieee-802-15-4-wireless/6lowpan.php> (accessed Apr. 17, 2022).
- [53] Preston. Gralla and Beatriz. Tarancón Alvaro, “Cómo funcionan las redes inalámbricas,” 2007, Accessed: Apr. 17, 2022. [Online]. Available:  
[https://books.google.com/books/about/C%C3%B3mo\\_funcionan\\_las\\_redes\\_inal%C3%A1mbricas.html?hl=es&id=-9G8AAAACAAJ](https://books.google.com/books/about/C%C3%B3mo_funcionan_las_redes_inal%C3%A1mbricas.html?hl=es&id=-9G8AAAACAAJ)





# Anexo I. Código fuente local.

```
/* =====  
 * == Código de funcionamiento local (no requiere conexión WIFI) ==  
 * =====  
 *  
 * Descripción:  
 * Toma datos de CO2 en ppm y muestra los colores en consecuencia  
 * Código válido para sensor con tira led de 18 LED's  
 * Autocalibración del SCD30 activada por defecto  
 * La alerta sonora viene activada  
 *  
 * =====  
 * == Versión 3.0 del código ==  
 * == Juan Boto, César Silva, ==  
 * == Simón López y Andrea Salgado ==  
 * == MediaLab, Marzo 2022 ==  
 * =====  
  
/* ===== CONFIGURACIÓN INICIAL ===== */  
  
// =====  
// == Librerías y defines necesarios ==  
// =====  
  
#include <EEPROM.h> // Memoria EEPROM para guardar  
valores no volatiles de configuración  
#include <Adafruit_SCD30.h> // Para sensor SCD30 (CO2, T y  
HR)  
#include <Adafruit_NeoPixel.h> // Para la tira LED  
#include "time.h" // Librería necesaria para  
controlar el tiempo de los procesos y usar la función millis  
  
#define EEPROM_SIZE 1 // Número de bits a los que queremos acceder  
#define PIN 19 // Pin de salida de datos del neopixel  
#define NUMPIXELS 18 // Número de leds utilizados (Poner 18 o 9  
dependiendo de la tira utilizada)  
  
// Llamada a la estructura de la librería de control del sensor  
Adafruit_SCD30 scd30;  
  
// Llamada a la estructura de la tira LED  
Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800); // Librería  
Adafruit_NeoPixel.h  
  
// 1. 18  
leds  
  
// 2. Pin  
salida datos tira LED (19)  
  
// 3. Tipo  
de LED NEO_GRB  
  
// 4.  
Version dispositivo 800 KHz
```

```
// =====  
// == Variables ==  
// =====  
  
// Variables de control  
  
int encender_led = 1; // Esta variable permite poner al neopixel en on/off  
bool buzzer_on = 0;  
// Esta variable permite poner al zumbador en on/off  
  
const char *CADENA = "cadena";  
const char *BUZZER = "buzzer";  
const char *MODO = "modo";  
  
const int total_tam = 5000, tam_res = 154;  
char total[total_tam];  
char temp[50];  
  
uint16_t dutyCycle_buzzer = 0;  
// con duty cycle 0, el zumbador está silenciado  
const int freq = 5000, ledChannel_buzzer = 0, resolution = 10, piezoPin =  
18; // frecuencia del zumbador, canal, resolución y output data pin  
  
// Variables de temporización  
// Usando la librería 'time' los intervalos enviados son controlados  
  
unsigned long currentTime = millis();  
unsigned long previousTime_0 = 0;  
const long interval_0 = 200;  
  
unsigned long previousTime_1 = 0;  
const long interval_1 = 2000;  
  
unsigned long previousTime_2 = 0;  
const long interval_2 = 1000 * 30;  
  
const int buzzer_sequence_tam = 6;  
int buzzer_sequence [buzzer_sequence_tam] = {500,500,0,500,0,500};  
  
// Variables auxiliares y de datos  
  
int aux = 0; // aux para pitido  
int sonar = 0; // sonar 0,1,2  
  
int co2 = 0, modo = 0, modo_antes=0; //co2: guarda el valor actual de co2,  
modo: cambia el color de los leds activando diferentes modos  
  
float media = 0, temper = 0, hum = 0;  
  
/* =====*/  
  
// =====  
// == Funciones ==  
// =====
```

```
/* ===== Función Modo y cambio de modo
=====*/
// Establece los colores para cada uno de los 3 modos de color: verde,
ambar y rojo;
// dependiendo de la cantidad de CO2 en el aire
// Si usamos tira de LED de 18 leds, NUMPIXELS = 18. Si usamos la de 9
leds, NUMPIXELS = 9 y
// entra en el else y ejecuta el código para los 9 LEDS

void cambio(int modo){

    if (NUMPIXELS == 18){

        int a_r[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        int a_g[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
        int a_b[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

        if (encender_led == 1) // Si la variable de control on/off esta en on(1)
        {
            switch (modo)
            {
                case 0: // Verde
                    a_g[12] = 255;
                    a_g[13] = 255;
                    a_g[14] = 255;
                    a_g[15] = 255;
                    a_g[16] = 255;
                    a_g[17] = 255;

                    break;

                case 1: // Ambar
                    a_g[6] = 165;
                    a_g[7] = 165;
                    a_g[8] = 165;
                    a_g[9] = 165;
                    a_g[10] = 165;
                    a_g[11] = 165;

                    a_r[6] = 255;
                    a_r[7] = 255;
                    a_r[8] = 255;
                    a_r[9] = 255;
                    a_r[10] = 255;
                    a_r[11] = 255;

                    break;

                case 2: //Rojo
                    a_r[0] = 255;
                    a_r[1] = 255;
                    a_r[2] = 255;
                    a_r[3] = 255;
                    a_r[4] = 255;
                    a_r[5] = 255;

                    break;
            }
        }
    }
}
```

```
        default:
            break;
    }
}

pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();

for (int i = 0; i < NUMPIXELS; i++)
{
    pixels.setPixelColor(i, pixels.Color(a_r[i], a_g[i], a_b[i]));
}

delay(1);
pixels.show(); }

else {

int a_r[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
int a_g[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};
int a_b[9] = {0, 0, 0, 0, 0, 0, 0, 0, 0};

if (encender_led == 1) //neopixels on are available
{
    switch (modo)
    {
        case 0: //verde
            a_g[6] = 255;
            a_g[7] = 255;
            a_g[8] = 255;

            break;
        case 1: //naraja
            a_g[3] = 165;
            a_g[4] = 165;
            a_g[5] = 165;

            a_r[3] = 255;
            a_r[4] = 255;
            a_r[5] = 255;

            break;
        case 2: //rojo
            a_r[0] = 255;
            a_r[1] = 255;
            a_r[2] = 255;

            break;
        default:
            break;
    }
}

pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();

for (int i = 0; i < NUMPIXELS; i++)
```

```
{
  pixels.setPixelColor(i, pixels.Color(a_r[i], a_g[i], a_b[i]));
}

delay(1);
pixels.show();

}
}

/* =====*/

/* ===== Función leer datos del SCD30
=====*/
// Lee los datos del SCD30 y actua en consecuencia. Cambia de modo y
comprueba el
// cambio de modo para emitir la alerta sonora.

void leer()
{
  if (scd30.dataReady()) {

    // Variables para guardar los datos del sensor
    co2 = scd30.CO2;
    hum = scd30.relative_humidity;
    temper = scd30.temperature;

    if (!scd30.read()){
      Serial.println("ERROR leyendo sensor...");
      return;
    }
    delay(500);

    if (co2 < 700)
      modo = 0;
    else if ((co2 >= 700) && (co2 < 1000))
      modo = 1;
    else
      modo = 2;

    if ((modo_antes == 0) && (modo == 1))
    {
      sonar = 1;
    }

    if ((modo_antes == 1) && (modo == 2))
    {
      sonar = 2;
    }

    if ((modo_antes == 0) && (modo == 2))
    {
      sonar = 2;
    }
  }
}
```

```
}

modo_antes = modo;
cambio(modo);
ledcWrite(ledChannel_buzzer, dutyCycle_buzzer);

Serial.print(co2);
Serial.print(",");
Serial.print(hum);
Serial.print(",");
Serial.print(temper);
Serial.print(",");
Serial.println();

}

}

/* =====*/

/* ===== Función emitir pitidos
=====*/
void pitido()
{
    if (sonar == 2) //varios pitidos
    {
        Serial.println("Emitiendo pitido doble... (co2>1000)");
        dutyCycle_buzzer = buzzer_sequence[aux];
        aux = aux + 1;
        if (aux >= buzzer_sequence_tam)
        {
            aux = 0;
            sonar = 0;
        }
    }
    else if (sonar == 1) //un pitido
    {
        Serial.println("Emitiendo un solo pitido... (co2>700)");
        dutyCycle_buzzer = 500;
        sonar = 0;
    }
    else dutyCycle_buzzer = 0;
    ledcWrite(ledChannel_buzzer, dutyCycle_buzzer);
}

/* =====*/

// =====
// == Programa principal ==
// =====

/* === CONFIGURACIÓN INICIAL === */

void setup() {
```

```
Serial.begin(115200);
pixels.begin();
Wire.begin();
EEPROM.begin(EEPROM_SIZE);

Serial.println("Iniciando prueba LED's");
cambio(0);
delay(200);
cambio(1);
delay(200);
cambio(2);
delay(200);
Serial.println("Prueba LED's completa");

buzzer_on = EEPROM.read(0); //Se lee el ultimo valor on/off del zumbador
antes de que el micro se apagara

ledcSetup(ledChannel_buzzer, freq, resolution); // Configuración de las
funcionalidades LED PWM
ledcAttachPin(piezoPin, ledChannel_buzzer); // Vincular el canal GPIO
para ser controlado

Serial.println("Iniciando el sensor SCD30...");
// Intentar inicializar SCD30
if (!scd30.begin()) {
  Serial.println("No se ha encontrado SCD30, revisar conexión");
  while (1) { delay(10); }
}

// CONFIGURACIÓN BÁSICA DEL SCD30

/* Ajustar tiempo entre medidas, rango 2-1800 segundos */
//scd30.setMeasurementInterval(2);
Serial.print("Intervalo de medidas: ");
Serial.print(scd30.getMeasurementInterval());
Serial.println(" segundos");

/* Señalar altura a la que se coloca el sensor respecto del nivel del mar
Gijón a unos 20m
*/
scd30.setAltitudeOffset(20);
Serial.print("Altura a nivel del mar: ");
Serial.print(scd30.getAltitudeOffset());
Serial.println(" metros");

Serial.println("\n\n");

/* Forzar calibración */
//scd30.forceRecalibrationWithReference(550);
Serial.print("Referencia de caliración: ");
Serial.print(scd30.getForcedCalibrationReference());
Serial.println(" ppm");
```



```
/* Activar ASC (Calibración automática)*/
scd30.selfCalibrationEnabled(false);
if (scd30.selfCalibrationEnabled()) {
    Serial.print("Calibración automática (ON)");}
else {
    Serial.print("Calibración automática (OFF)");
}

Serial.println("\n\n");

Serial.println("Configuración inicial COMPLETA CON EXITO");
Serial.println("\n\n");

}

/* === BUCLE PRINCIPAL === */

void loop()
{

    currentTime = millis();

    if (currentTime - previousTime_0 > interval_0) //cada 200 ms
    {
        previousTime_0 = currentTime;
        pitido();
    }

    if (currentTime - previousTime_1 > interval_1) //cada 2 s
    {
        previousTime_1 = currentTime;
        leer();
    }

    if (currentTime - previousTime_2 > interval_2) //cada 15 min
    {

        previousTime_2 = currentTime;

    }

}

/* =====*/
```

# Anexo II. Código fuente Wifi.

## Archivo .c

```
/*
=====
=====
* == Código que envía datos de un sensor a la base de datos de MediaLab en
IONOS ==
*
=====
=====
*
* Descripción:
* Toma datos de CO2 en ppm, temperatura en °C y humedad relativa en %
* Código válido para sensor v2.0 (pcb v1.0) con tira led de 18 LED's
* La alerta sonora viene activada en este caso, ASC desactivado
*
* =====
* == Versión 5.0 del código ==
* == César Silva y Juan Boto ==
* == MediaLab, 2022 ==
* =====
*/

// Librerías de sensor y leds
#include <Adafruit_SCD30.h> // Para sensor SCD30 (CO2, T y HR)
#include <Adafruit_NeoPixel.h> // Para la tira LED

// Librerías watchdog
#include <esp_task_wdt.h>

// Librerías Wifi y TCP si se utiliza ESP32
#include <WiFi.h>
#include <AsyncTCP.h>

// #include <ESP8266WiFi.h>
// #include <ESPAsyncTCP.h>

// Librerías sincronización y cliente HTTP
#include <ESPAsyncWebServer.h>
#include <HTTPClient.h>

// Librería para calcular la mediana de las medidas tomadas
#include "QuickMedianLib.h"

// Archivo header con las credenciales SSID y password de la red Wifi
#include "credentials.h"

// Defines con parametros constantes
#define PIN 19 // pin de salida de control neopixel
#define NUMPIXELS 18 // número de píxeles
#define WDT_TIMEOUT 200 // Temporizador Watchdog
```

```
Adafruit_SCD30 scd30;

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);

HTTPClient http;

// variables para el cálculo de la mediana de medidas tomadas en 30
segundos
int valoresMediana[15] = {500, 500, 500, 500, 500, 500, 500, 500, 500, 500,
500, 500, 500, 500, 500};
int valoresMedianaLength = sizeof(valoresMediana) / sizeof(int);
int iteracionMediana = 0;
int mediana;

const int small_delay = 50;
int wifiSetupAttemp = 0;

int encender_led = 1; // Esta variable permite poner al neopixel en on/off
int buzzer_on = 1;    // Esta variable permite poner al zumbador en on/off

// Variables de temporización

unsigned long previousTime_0 = 0;
const unsigned long interval_0 = 200; // 200 ms

unsigned long previousTime_1 = 0;
const unsigned long interval_1 = 2000; // 2 segundos

unsigned long previousTime_2 = 0;
const unsigned long interval_2 = 140000; // 140 segundos

// Variables auxiliares, modo, sonar...

int sonar = 0; // sonar 0,1,2

int co2 = 0, modo = 0; // co2: guarda el valor actual de co2, modo: cambia
el color de los leds
int co2_ant = 0, modo_antes = 0; // Guarda el estado anterior de las
variables

int hum = 0;
float temper = 0;

int contadorMedianas = 0; // Contador de medianas tomadas

int contadorPitidos1 = 0; // un pitido
int contadorPitidos2 = 0; // pitido doble

int emisor = 18; // definir el pin digital del Arduino

/*
 * =====
 * ==          FUNCIONES          ==
 * =====
 */
```

```
*/  
  
/* == CAMBIAR COLOR LEDS ==*/  
// Cambia el color dependiendo de 'modo': verde, ambar y rojo; dependiendo  
de la cantidad de CO2 en el aire  
// Hay dos modos extra, azul para conexión wifi si se desconecta y rojo en  
zigzag por si no se lee  
// el sensor.  
  
void change_led_colors(int modo)  
{  
  
    // Declaracion array color leds  
    int a_r[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
    int a_g[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
    int a_b[18] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};  
  
    if (encender_led == 1) // Si la variable de control on/off esta en on(1)  
    {  
  
        switch (modo)  
        {  
            case 0: // Verde  
  
                a_g[12] = 255;  
                a_g[13] = 255;  
                a_g[14] = 255;  
                a_g[15] = 255;  
                a_g[16] = 255;  
                a_g[17] = 255;  
  
                break;  
  
            case 1: // Ambar  
  
                a_g[6] = 165;  
                a_g[7] = 165;  
                a_g[8] = 165;  
                a_g[9] = 165;  
                a_g[10] = 165;  
                a_g[11] = 165;  
  
                a_r[6] = 255;  
                a_r[7] = 255;  
                a_r[8] = 255;  
                a_r[9] = 255;  
                a_r[10] = 255;  
                a_r[11] = 255;  
  
                break;  
  
            case 2: // Rojo  
  
                a_r[0] = 255;  
                a_r[1] = 255;  
                a_r[2] = 255;  
                a_r[3] = 255;
```

```
a_r[4] = 255;
a_r[5] = 255;

break;

case 3: // Azul

a_b[0] = 255;
a_r[1] = 255;
a_b[2] = 255;
a_b[3] = 255;
a_r[4] = 255;
a_b[5] = 255;

break;

case 4: // rojo zigzag

a_r[0] = 255;
a_r[6] = 255;
a_r[12] = 255;

break;

default:
break;
}

pixels.setPixelColor(0, pixels.Color(0, 0, 0));
pixels.show();

for (int i = 0; i < 18; i++)
{
pixels.setPixelColor(i, pixels.Color(a_r[i], a_g[i], a_b[i]));
}

pixels.show();
}
}

/* == CONEXIÓN A WIFI ==*/
// Función que conecta a una red Wifi dandole las credenciales, que se
encontraran en credentials.h

void wifiSetup()
{
WiFi.begin(WIFI_SSID, WIFI_PASS);

while (WiFi.status() != WL_CONNECTED)
{
delay(500);
wifiSetupAttemp++;
Serial.println("Conectando al WiFi...");
if (wifiSetupAttemp >= 20)
{
wifiSetupAttemp = 0;
Serial.println("Conectando al WiFi...");
}
}
}
}
```

```
        change_led_colors(3);
        break;
    }
}

if (WiFi.status() != WL_CONNECTED)
{
    Serial.println("No se ha podido conectar al Wifi, se intentará de nuevo
en unos segundos");
}
else
{
    Serial.println(";Conectado a Wifi!");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}
}

/* == LEER DATOS DEL SENSOR ==*/
// Función que lee los datos del sensor SCD30, mete los valores en el array
de la mediana
// cambia de modo (color) en consecuencia y activa el pitido si ha cambiado
de modo y procede.

void leer()
{
    co2_ant = co2;

    if (scd30.dataReady())
    {
        co2 = scd30.CO2;
        valoresMediana[iteracionMediana] = co2;
        iteracionMediana++;
        hum = scd30.relative_humidity;
        temper = scd30.temperature;

        if (!scd30.read())
        {
            Serial.println("ERROR leyendo sensor...");
            return;
        }
        delay(500);

        if (co2 < 700)
            modo = 0;
        else if ((co2 >= 700) && (co2 < 1000))
            modo = 1;
        else
            modo = 2;

        if ((modo_antes == 0) && (modo == 1))
            sonar = 1;

        if ((modo_antes == 1) && (modo == 2))
            sonar = 2;

        if ((modo_antes == 0) && (modo == 2))
```

```
        sonar = 2;

modo_antes = modo;

change_led_colors(modo);

if (iteracionMediana >= 15) // cuando he guardado 15 datos
    almacenar();

Serial.print(co2);
Serial.print(",");
Serial.print(hum);
Serial.print(",");
Serial.print(temper);
Serial.print(",");
Serial.print(WiFi.macAddress());
Serial.print(",");
Serial.println();
//Serial.print(modo);
//Serial.print(",");
//Serial.print(co2_ant);
//Serial.println();
    }
}

/* == CALCULAE MEDIANA Y SUBIR VALOR A LA BASE DE DATOS ==*/
// Se ejecuta al llegar a 15 valores guardados en el array y calcula la
mediana de los 15 valores
// Encía el valor de la mediana y comprueba que haya llegado a la base de
datos
// Por otro lado, si dos medianas son menores que 360 ppm durante 360 s,
recalibra el sensor

void almacenar()
{

    delay(small_delay);

    mediana = QuickMedian<int>::GetMedian(valoresMediana,
valoresMedianaLength);

    String serverPath = "https://www.medialab-
uniovi.es/post_sensorCO2.php?MAC_sensor='" + WiFi.macAddress() + "'&co2=" +
mediana + "&humedad=" + hum + "&temp=" + temper;
    http.begin(serverPath.c_str());
    int httpResponseCode = http.GET();

    if (httpResponseCode > 0)
    {
        Serial.print("HTTP Response code: ");
        Serial.println(httpResponseCode);
        String payload = http.getString();
        Serial.println(payload);
    }
    else
    {
        Serial.print("Error code: ");
```

```
Serial.println(httpResponseCode);
Serial.print(" Descripción: Error subiendo datos a la base de datos ");
Serial.println();
}
http.end();

delay(small_delay);

iteracionMediana = 0;

if (mediana < 360)
{
    contadorMedianas++;
}

if (contadorMedianas == 2)
{
    delay(2000);
    contadorMedianas = 0;
    Serial.println("Forzando calibración debido a datos co2 < 360 ppm");
    scd30.forceRecalibrationWithReference(415);
    delay(2000);
    Serial.println("Comprobando calibración automática: ");
    if (scd30.selfCalibrationEnabled())
    {
        Serial.print("Calibración automática (ON)");
    }
    else
    {
        Serial.print("Calibración automática (OFF)");
    }
}
}

/* == EMITIR PITIDO ==*/
// Si el zumbador está activo, se emite pitido o pitido doble al bajar la
calidad de aire
// Solo puede pitar dos veces por cada 140 segundos para no molestar
demasiado

void pitido()
{
    if (buzzer_on == 1)
    {
        if (sonar == 2 && contadorPitidos2 <= 1) // varios pitidos
        {
            contadorPitidos2++;
            pinMode(emisor, OUTPUT); // definir el pin como salida
            for (int i = 0; i < 70; i++) { // emitir un sonido
                digitalWrite(emisor, HIGH); // salida en ALTO
                delay(1); // espera 1 ms
                digitalWrite(emisor, LOW); // salida en BAJO
                delay(1);
            }
            delay(50);
        }
    }
}
```



```
pinMode(emisor, OUTPUT); // definir el pin como salida
for (int i = 0; i < 70; i++) { // emitir un sonido
digitalWrite(emisor, HIGH); // salida en ALTO
delay(1); // espera 1 ms
digitalWrite(emisor, LOW); // salida en BAJO
delay(1);
}
digitalWrite(emisor, LOW); // salida en BAJO
sonar=0;
}

else if (sonar == 1 && contadorPitidos1 <= 1) // un pitido
{
    contadorPitidos1++;
    pinMode(emisor, OUTPUT); // definir el pin como salida
    for (int i = 0; i < 70; i++) { // emitir un sonido
digitalWrite(emisor, HIGH); // salida en ALTO
delay(1); // espera 1 ms
digitalWrite(emisor, LOW); // salida en BAJO
delay(1);
}
digitalWrite(emisor, LOW); // salida en BAJO
sonar=0;
}
}
}

/* =====*/

/*
* =====
* ==      CONF. INICIAL      ==
* =====
*/

void setup ()
{
    // CONFIGURACIÓN SERIAL, WIFI, LEDES

    Serial.begin(115200);
    wifiSetup();

    Serial.println("Configurando Watchdog...");
    esp_task_wdt_init(WDT_TIMEOUT, true); // enable panic so ESP32 restarts
    esp_task_wdt_add(NULL); // add current thread to WDT watch
    Serial.println("Watchdog configurado con éxito");

    pixels.begin();
    pixels.setBrightness(100);
    //Wire.begin();

    delay(small_delay);

    Serial.println("Iniciando prueba LED's");
    change_led_colors(0); // initial led colors
```

```
delay(500);
change_led_colors(1);
delay(500);
change_led_colors(2);
delay(1000);
change_led_colors(0);
delay(200);
Serial.println("Prueba LED's completa");

// INICIO SENSOR SCD30

Serial.println("Iniciando el sensor SCD30...");
scd30.begin();
// Intentar inicializar SCD30
if (!scd30.begin())
{
  Serial.println("No se ha encontrado SCD30, revisar conexión");
  change_led_colors(4);
  while (1)
  {
    delay(3000);
    scd30.begin();
    delay(1000);
    if (scd30.begin() == true) {
      Serial.print("SCD30 Iniciado. ");
      delay(1000);
      break;
    }
  }
}

if (scd30.begin() == true)
  Serial.print("SCD30 Iniciado. ");

// CONFIGURACIÓN BÁSICA DEL SCD30

/* Ajustar tiempo entre medidas, rango 2-1800 segundos */
// scd30.setMeasurementInterval(2);
Serial.print("Intervalo de medidas: ");
Serial.print(scd30.getMeasurementInterval());
Serial.println(" segundos");

/* Señalar altura a la que se coloca el sensor respecto del nivel del mar
Gijón a unos 20m
*/
scd30.setAltitudeOffset(20);
Serial.print("Altura a nivel del mar: ");
Serial.print(scd30.getAltitudeOffset());
Serial.println(" metros");

Serial.println("\n\n");

/* Forzar calibración */
```

```
//scd30.forceRecalibrationWithReference(750);
Serial.print("Referencia de caliración: ");
Serial.print(scd30.getForcedCalibrationReference());
Serial.println(" ppm");

/* Activar ASC (Calibración automática)*/
scd30.selfCalibrationEnabled(false);

if (scd30.selfCalibrationEnabled())
    Serial.print("Calibración automática (ON)");
else
    Serial.print("Calibración automática (OFF)");

scd30.setTemperatureOffset(500);

// REINICIO DEL WATCHDOG
esp_task_wdt_reset();
Serial.println("Watchdog puesto a 0.");

Serial.println("\n\n");
Serial.println("Configuración inicial COMPLETA CON EXITO");
Serial.println("\n\n");
}

/* =====*/

/*
 * =====
 * ==          BUCLE PRINCIPAL          ==
 * =====
 */

void loop()
{

    if (millis() - previousTime_0 >= interval_0) // cada 200 ms
    {
        previousTime_0 = millis();
        pitido();
    }

    if (millis() - previousTime_1 >= interval_1) // cada 2 seg
    {
        previousTime_1 = millis();
        leer();
        if (co2 != co2_ant)
            esp_task_wdt_reset();
        delay(small_delay);
    }

    if (millis() - previousTime_2 >= interval_2) // cada 140 seg
    {
        previousTime_2 = millis();
        if (WiFi.status() != WL_CONNECTED)
```

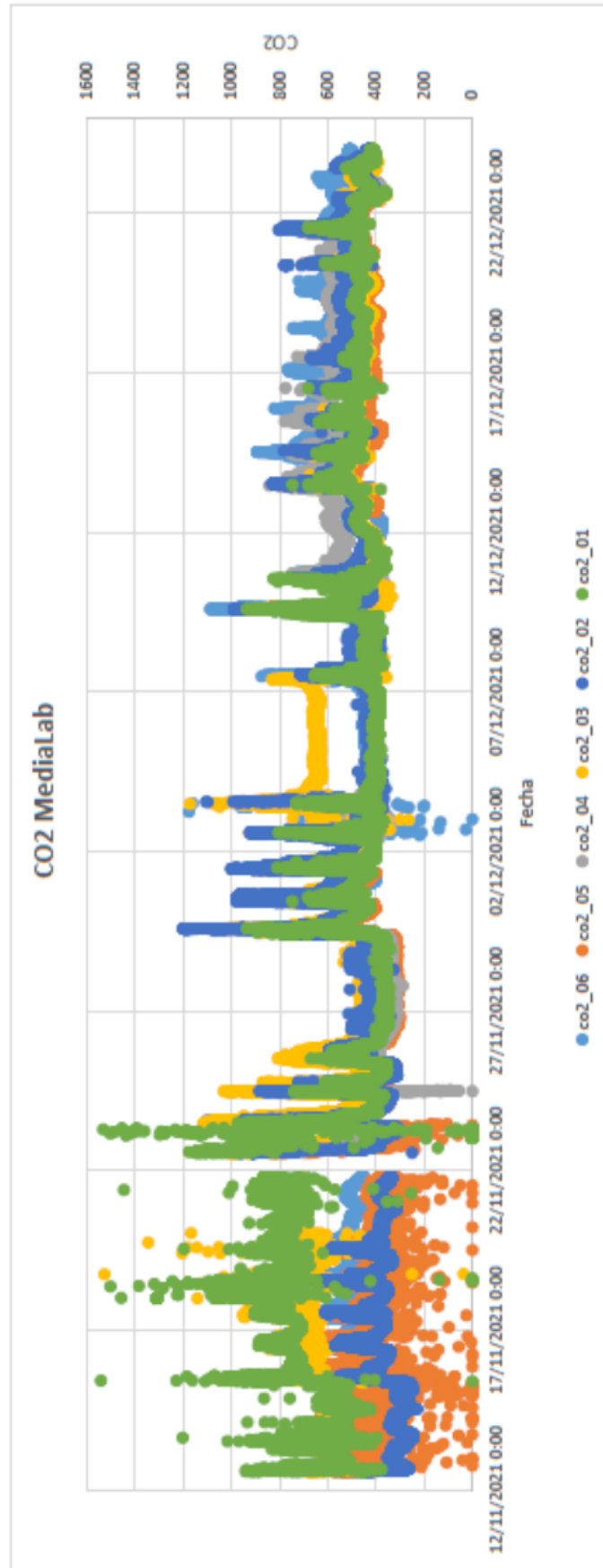
```
{
    wifiSetup();
}
contadorPitidos1 = 0;
contadorPitidos2 = 0;
contadorMedianas = 0;
}
}
/* =====*/
```

Archivo .h

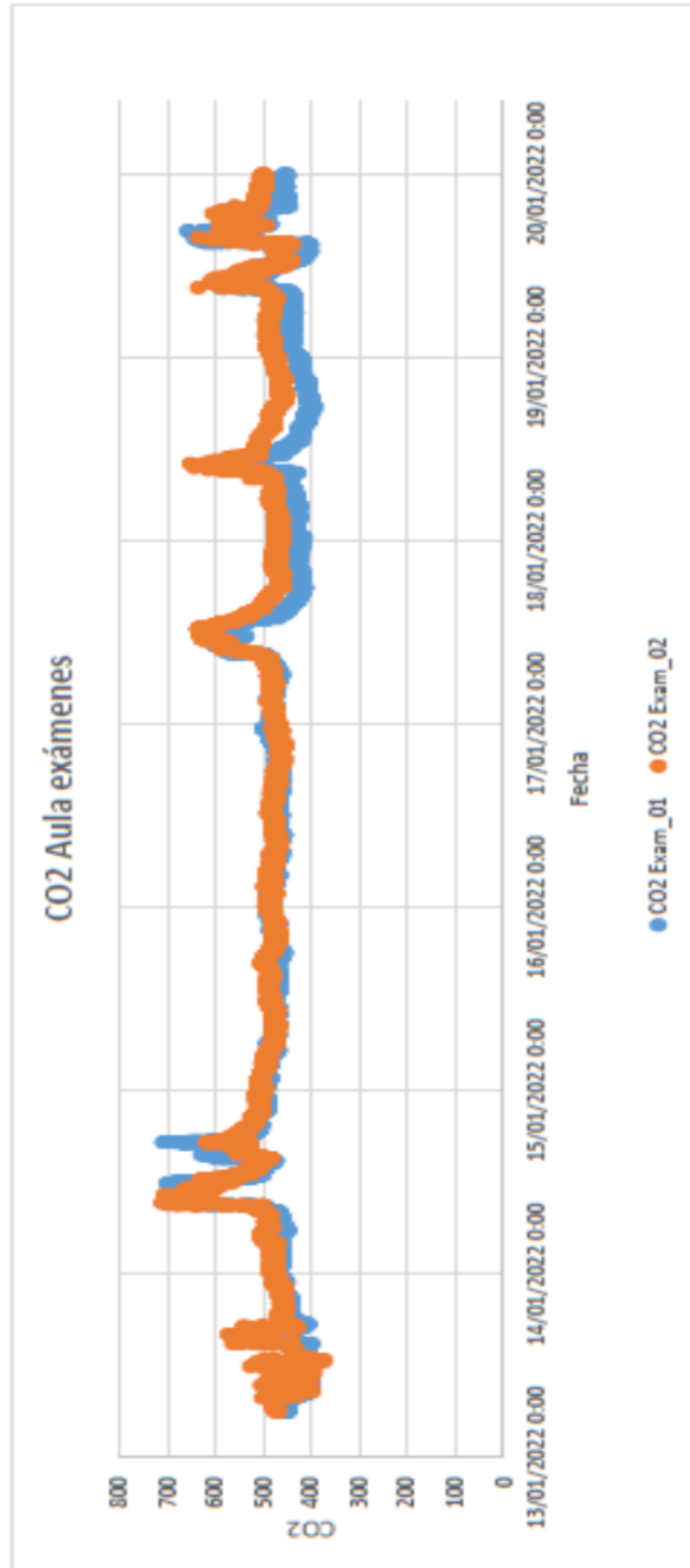
```
#define WIFI_SSID ""  
#define WIFI_PASS ""
```

# Anexo III. Gráficas.

(Gráficas en siguiente página)



Gráfica 1.- Datos obtenidos del experimento en espacio MediaLab.



Gráfica 2.- Datos obtenidos del experimento en aula de exámenes.