

TRABAJO DE GRADO
Opción Investigación o Proyecto de Grado

**Medición y captura de dióxido de carbono (CO₂) mediante el uso de sensores
ambientales**

Corporación Universitaria Remington.
Facultad de Ingeniería.
Ingeniería Ambiental.

Luz Mirian Castañeda Portela.
Tutor: Juan Esteban Velez Alvarez.
Proyecto de grado.
2026.

Tabla de Contenidos

Resumen.....	5
Palabras clave.....	5
Introducción	6
Métodos de captura de CO ₂	8
Adsorción con óxidos metálicos.	8
Precipitación de carbonatos o carbonatación mineral.	8
Membranas porosas.	9
Reactor de columna de burbujeo.	9
Planteamiento del problema.....	13
Objetivos	14
Objetivo General	14
Objetivos específicos	14
Metodología	15
Diseño del sensor de resonancia de Helmholtz.	15
Validación del modelo del resonador.	16
Diseño y construcción del reactor de columna de burbujeo.	18
Generación de CO₂.	18
Integración y diseño experimental.	19
Prueba con burbujeo.	19
Prueba sin burbujeo.	20
Adquisición y procesamiento de datos.	21
Resultados y Discusión	22
Análisis estadístico.....	22
Prueba con burbujeo.	23
Prueba sin burbujeo.	25
Comparación entre pruebas.	27
Dinámica de respuesta del sensor.	28
Conclusiones	30
Referencias.....	31
Anexos	33
Anexo A. Desarrollo numérico de la validación del modelo del resonador.	33
Anexo B. Código del microcontrolador (Arduino Due).	33
Anexo C. Código de monitoreo y graficación (Python).	36
Anexo D. Origen de las tablas y las figuras.	42

Lista de tablas

Tabla 1. Estadísticos descriptivos de la frecuencia de resonancia f_0 en cada configuración.	22
Tabla 2. Datos registrados en la prueba con burbujeo.	24
Tabla 3. Datos registrados en la prueba sin burbujeo.	26

Lista de figuras

Figura 1. Modelo CAD del resonador de Helmholtz (cavidad cilíndrica con cuello tubular).....	16
Figura 2. Montaje del sensor de resonancia de Helmholtz con la placa Arduino Due.	18
Figura 3. Sistema de generación de CO ₂ mediante la reacción de bicarbonato de sodio con ácido acético conectado al reactor de columna de burbujeo.....	19
Figura 4. Montaje de la prueba con burbujeo: la manguera de entrada se conecta hasta el fondo del reactor.	20
Figura 5. Montaje de la prueba sin burbujeo: la manguera de entrada no entra en contacto con la solución.	20
Figura 6. Evolución temporal de la frecuencia de resonancia f_0 en la prueba con burbujeo.	23
Figura 7. Evolución temporal de la frecuencia de resonancia f_0 en la prueba sin burbujeo.	25
Figura 8. Comparación de la frecuencia de resonancia f_0 en ambas pruebas frente a las líneas esperadas para 1 % y 5 % de CO ₂	27

Resumen

En este trabajo, diseñamos e implementamos un sistema integrado de monitoreo y captura de CO₂ en tiempo real en un modelo a escala (prototipo) de una planta de producción de hidrógeno esmeralda mediante reformado de metano. Dado que esta es la fuente industrial más importante de hidrógeno y el CO₂ se libera como subproducto, proponemos medir y retener este gas en un único sistema asequible y de bajo costo. Monitoreamos este gas con un sensor basado en la resonancia de Helmholtz. La señal se desplazó con la concentración de CO₂ y utilizamos una placa Arduino Due y Python para capturarla. Elegimos capturarlo en un reactor de columna de burbujeo con hidróxido de sodio, ya que hemos estudiado la adsorción con óxidos metálicos, la precipitación con carbonato y membranas porosas. Hemos realizado dos pruebas, con y sin burbujeo, para producir CO₂ a partir de bicarbonato de sodio y ácido acético como fuente de laboratorio sustituta. El sensor ha mostrado la evolución del CO₂ en tiempo real, en ambas configuraciones, con una respuesta más rápida pero más dispersa cuando se forzó el burbujeo (coeficiente de variación de 10.5 %) frente a una respuesta más estable sin burbujeo (5.3 %), y una validación del modelo del resonador con un error relativo del 19.2 % frente a la velocidad del sonido teórica.

Palabras clave

Captura de CO₂; reformado de metano; hidrógeno esmeralda; resonancia de Helmholtz; reactor de columna de burbujeo.

Introducción

La quema de combustibles fósiles sigue siendo una de las causas principales del cambio climático y las cifras lo confirman, en 2023 el mercado energético estuvo dominado por combustibles fósiles, que emitieron alrededor de 36.8 mil millones de toneladas de CO₂ a la atmósfera. Pero el panorama está cambiando y la transición hacia energías renovables avanza rápidamente (Anand, 2025). El hidrógeno ha ganado relevancia en la descarbonización como una alternativa sostenible a los combustibles fósiles en áreas donde la electrificación es un desafío (industria pesada, transporte, generación de energía, almacenamiento a largo plazo) (Younus, 2025). Hay un enorme potencial para esto, en los modelos del sistema energético europeo de 2050, la electricidad y el hidrógeno podrían representar aproximadamente el 60% y el 6% del consumo total (van der Zwaan, 2025). Sin embargo, si el hidrógeno es realmente limpio depende de cómo se produzca, por lo que el hidrógeno verde generado a partir de fuentes renovables, ha sido promovido como una parte vital de la transición hacia una economía baja en carbono (Jumah, 2024).

El problema principal es la fuente de hidrógeno que tenemos hoy en día. La mayor parte de este hidrógeno se crea a través de la tecnología de reformado de metano con vapor, que ha sido ampliamente adoptada pero también tiene sus propios inconvenientes. Este hidrógeno tiene emisiones que no pueden ser totalmente eliminadas, por lo que se utiliza la captura y almacenamiento de carbono para reducir su efecto (Cho, 2022). Cuando ese hidrógeno proviene de recursos fósiles y libera CO₂ a la atmósfera, se define como hidrógeno gris (Navas-Anguita, 2021). Como resultado, surge una alternativa que

no enfrenta este problema, el hidrógeno esmeralda, en el cual el metano producido ya no es fósil sino renovable, como el biogás. Un estudio examinó una planta que convierte biogás en hidrógeno esmeralda de emisión negativa a través del reformado electrificado y la separación de CO₂. Un aspecto importante de este enfoque es que tiene un doble beneficio desde la perspectiva técnica, electrificar el reformador conduce a más gas de síntesis y permite reactores más compactos y flexibilidad operativa al evitar la combustión y la transferencia de calor a través de las paredes bajo presión. También es económicamente viable, con un contenido de energía renovable del 95%. Para este trabajo, se entiende que el hidrógeno esmeralda es hidrógeno a partir de biogás producido a partir del reformado de metano electrificado, que captura y separa el CO₂ producido (Nava, 2024). La diferencia con el hidrógeno gris producido a partir de gas natural fósil que libera CO₂ a la atmósfera, está precisamente ahí; en este caso, el metano es renovable y el dióxido de carbono no se libera, sino que se captura y separa. Esto reduce en gran medida la huella de carbono y en algunos casos, incluso se pueden lograr emisiones negativas.

Y es precisamente el reformado de metano con separación de CO₂ lo que motiva este trabajo. A diferencia del gris, el enfoque esmeralda introduce sistemas de captura y separación que evitan que el CO₂ del proceso de reformado llegue directamente a la atmósfera, de modo que este gas deje de ser un subproducto contaminante y se convierta en un flujo controlado que debe ser medido y capturado. Pero llevar esta idea a una planta real requiere primero verificar bajo condiciones controladas y que la etapa de medición y captura esté funcionando. Ese es el objetivo de este proyecto, un modelo a escala de

laboratorio para la combinación de monitoreo en tiempo real y captura de CO₂ asociado con la producción de hidrógeno esmeralda mediante reformado de metano. El prototipo no pretende reproducir toda la planta, se trata solo de su etapa de CO₂, utiliza una fuente de gas sustituta y es una prueba de concepto accesible y de bajo costo, que se necesita para demostrar y caracterizar esta etapa para ser considerada de baja o cero emisiones.

Métodos de captura de CO₂

Antes de construir el sistema, era necesario decidir cómo capturar CO₂, por lo que se revisaron las ventajas y limitaciones de cuatro alternativas.

Adsorción con óxidos metálicos. Donde el CO₂ se fija a sólidos simples como el óxido de calcio o magnesio. La principal ventaja de esto sobre las soluciones de aminas es que no presenta los problemas más serios de estas, es decir, el alto consumo de energía para la regeneración, la corrosión del equipo y el costo del sorbente, que lo hacen una opción prometedora para el futuro (Kumar, 2014). Pero este depende de las condiciones de operación, y la capacidad de adsorción de algunos de los carbones que se modifican con óxidos metálicos no es alta (Sun, 2015). Además de eso, hay compromisos térmicos, ya que el MgO es más barato y se regenera entre 450 y 500 °C, pero los sorbentes de CaO necesitan llegar a 700 °C (Correia, 2023).

Precipitación de carbonatos o carbonatación mineral. En este método el CO₂ reacciona con cationes de calcio o magnesio para formar carbonatos sólidos estables. En reactores continuos, un gas rico en CO₂ se pasa a través de un alimento acuoso rico en calcio, con el gas siendo absorbido en un proceso selectivo y precipitado como vaterita o calcita (Tiefenthaler, 2022). Una ventaja importante es la permanencia, estos carbonatos

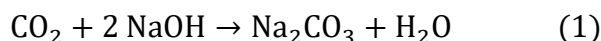
pueden ser estables a lo largo de escalas de tiempo geológicas y, por lo tanto, pueden almacenarse durante mucho tiempo (Nagireddi, 2024). El inconveniente es el costo, ya que consume mucha energía y el procesamiento de minerales es costoso; incluso si tiene sentido usar desechos industriales como materia prima, es difícil usarlo comercialmente debido a su baja reactividad y al costo (Hamed, 2025).

Membranas porosas. Estas membranas separan el CO_2 aprovechando diferencias de tamaño molecular, permeabilidad y solubilidad. Tienen puntos fuertes importantes, buena eficiencia, equipos y procesos sencillos y bajo consumo energético (Xu, 2021), y los materiales más avanzados, como los polímeros porosos orgánicos, ofrecen gran área superficial, alta permeabilidad, poros de tamaño ajustable y una procesabilidad superior a la de materiales clásicos como las zeolitas o la alúmina (Cao, 2024). Su principal limitación es conocida, hoy las membranas chocan con el compromiso entre permeabilidad y selectividad, y muchas se plastifican o se degradan físicamente cuando trabajan con presión y humedad altas (Lopez, 2026).

Reactor de columna de burbujeo. En este método el gas se inyecta en forma de burbujas a través de una solución alcalina. Los ensayos muestran que, cuanto mayor es la concentración de NaOH en el líquido y más alta la de CO_2 en el gas, mejor es la eficiencia de remoción (Michelson, 2024), algo que confirman otros estudios en los que la velocidad de reacción y la captura dependen en buena medida de la concentración de NaOH dentro del rango en que se produce Na_2CO_3 (Fleischer, 1996). Y rinde bien frente a otros álcalis: para MEA (Monoetanolamina), NaOH y KOH a 0,25 M, las eficiencias

máximas de remoción superaron el 60 % y crecieron al aumentar la concentración del solvente (Gul, 2023).

Al comparar estas cuatro técnicas, el reactor de columna de burbujeo con hidróxido de sodio ofrece el mejor equilibrio entre eficiencia, simplicidad y costo, y resulta ideal para un prototipo de laboratorio. De hecho, entre los solventes, el NaOH es el que elimina más CO₂ con un 92 a 99% (Sa'adiyah, 2021) y se ha encontrado que captura más del 95% en pruebas con bicarbonato de sodio de pureza superior al 94% (Chan, 2025). En comparación con las aminas, puede ser más económico para la misma cantidad de absorción, con una absorción máxima del 95% al 5% en peso, y el MEA requiere un 15% para ser igual de efectivo (Ameri, 2025). La reacción en la neutralización del CO₂ (que es un óxido ácido) con NaOH, una base fuerte, produce Na₂CO₃ y agua (H₂O), ecuación (1).



Este proceso no solo permite retener el CO₂, sino también su fijación química en forma de carbonato que ayuda a capturar y evitar su liberación al medio ambiente. Por lo tanto, es una alternativa accesible, eficiente y adecuada al montaje experimental propuesto.

Para determinar cuánto CO₂ se ha capturado es necesario medir, y aquí interviene el sensor basado en la resonancia de Helmholtz. El principio es sencillo, la frecuencia de resonancia depende de la velocidad del sonido, que a su vez depende de la masa molar del gas, y se puede usar un resonador piezoeléctrico simple para determinar la composición de mezclas binarias como CO₂, CH₄ o H₂ en el aire (Keeratrirawee, 2022). El método llega a ser muy sensible, la espectroscopía fotoacústica de un resonador de

Helmholtz diferencial resulta en un límite de detección equivalente al ruido de 160 ppmv de CO₂ (Alahmari, 2019), y trabajos recientes sugieren que su sensibilidad para gases como el CO₂ aumenta con la geometría del resonador al cambiar la forma del resonador (Zaky, 2025). Además, medir la frecuencia proporciona una ventaja práctica, estos sensores pueden cubrir un amplio rango sin perder velocidad de respuesta (Berndt, 2022), y la frecuencia se lee directamente como una señal, se necesita muy poca latencia (Suchenek, 2018), justo lo que se necesita para el monitoreo en tiempo real.

Finalmente, la captura y el procesamiento de datos se realizan con herramientas de código abierto, la placa Arduino Due toma la señal y Python la procesa y grafica. Tal combinación ha sido bien establecida en instrumentos científicos de bajo costo; existen sistemas con software abierto en Python que pueden tomar datos de cualquier PC con una simple conexión USB (Grinias, 2016). En la medición específica de CO₂, los sistemas con Arduino son baratos y fáciles de adaptar a otras necesidades (Pino, 2019), y estas plataformas económicas registran la variabilidad de CO₂ con buena resolución, superando las barreras de costo y funcionalidad de los analizadores comerciales (Blackstock, 2019). Los programas toman los datos a intervalos regulares y los envían para su visualización en tiempo real (Reivanth, 2024), cerrando el ciclo de captura y monitoreo de CO₂ que articula este proyecto.

Con todo esto como punto de partida, el documento plantea primero el problema del CO₂ que se libera al producir hidrógeno por reformado de metano y la necesidad de vigilarlo de forma continua; luego fija el objetivo de reunir medición y captura en un mismo sistema de bajo costo; y a continuación describe la metodología el diseño del

sensor de Helmholtz, la construcción del reactor de columna de burbujeo y la generación controlada de CO_2 para terminar evaluando su desempeño con dos pruebas experimentales, con y sin burbujeo.

Planteamiento del problema.

La producción de hidrógeno mediante el reformado de metano genera CO_2 como uno de sus principales subproductos. Esta es la ruta más común a escala industrial y el problema es aún peor aquí, una tecnología diseñada para proporcionar un combustible limpio eventualmente se convierte en una fuente significativa de emisiones y, por lo tanto, el hidrógeno producido de esta manera se clasifica como gris. El hidrógeno esmeralda a partir del reformado de metano renovable con captura de CO_2 solo podría realizarse si se lleva a cabo y se verifica. En muchos casos, sin embargo, la captura no es continua y tiene baja eficiencia para la captura. Como la captura de CO_2 no es continua ni eficiente en todos los casos, puede liberarse al aire y las emisiones de gases de efecto invernadero son mayores debido a esto. Por lo tanto, el problema que motivó este trabajo es monitorear continuamente y, al mismo tiempo, capturar eficientemente el CO_2 generado durante el reformado de metano, a través de un sistema accesible y de bajo costo que permita evaluar el rendimiento.

Justificación: Validar la etapa de captura y medición del CO_2 es precisamente lo que distingue al hidrógeno esmeralda del gris, sin esa etapa funcionando y cuantificada, el reformado de metano seguiría liberando su CO_2 a la atmósfera. Por eso, demostrar esta etapa a escala de prototipo es un paso necesario hacia una producción de hidrógeno de baja o nula emisión. El proyecto pretende resolver esta problemática integrando la medición y la captura del CO_2 en un único sistema. La concentración del gas se monitorea en tiempo real con un sensor basado en la resonancia de Helmholtz, conectado a una placa Arduino Due y procesado en Python, mientras que la captura se realiza en un reactor de columna de burbujeo con hidróxido de sodio. Dado que el objetivo es validar esta etapa y no reproducir una planta completa, el CO_2 se genera con una fuente sustituta de laboratorio en lugar de un reformador real. De este modo, en lugar de liberarse, el CO_2 generado se mide y se retiene, lo que permite cuantificar el desempeño del sistema y ofrecer una alternativa de bajo costo y replicable para reducir las emisiones asociadas a la producción de hidrógeno.

Objetivos

Objetivo General

Diseñar e implementar un sistema integrado de monitoreo en tiempo real y captura de CO₂, aplicado a la etapa de captura y medición del dióxido de carbono asociada a la producción de hidrógeno esmeralda mediante reformado de metano, y validarlo mediante un prototipo experimental a escala de laboratorio.

Objetivos específicos

- Configurar un sensor basado en el principio de resonancia de Helmholtz, integrado con una placa Arduino Due, para la medición continua y en tiempo real de la frecuencia de resonancia f_0 del sistema, asociada a la concentración de CO₂ presente en el gas.
- Desarrollar un programa en Python que permita la adquisición y visualización gráfica en tiempo real de la frecuencia de resonancia f_0 registrada por el sensor.
- Diseñar y construir un sistema de captura de CO₂ basado en un reactor de columna de burbujeo que utilice una solución de hidróxido de sodio (NaOH) como medio absorbente.
- Validar el funcionamiento integrado de los sistemas de monitoreo y captura mediante pruebas experimentales de laboratorio con datos reales del sensor, evaluando su desempeño y discutiendo su potencial de aplicación en procesos de mayor escala.

Metodología

El trabajo se desarrolló bajo un enfoque experimental de carácter comparativo. El montaje se construyó por etapas, en el orden en que se abordó el desarrollo, primero el sistema de monitoreo basado en un sensor de resonancia de Helmholtz, luego el reactor de columna de burbujeo para la captura, y finalmente la integración de ambos con la fuente de generación de CO₂ para realizar las pruebas. Conviene precisar que el montaje es un prototipo a escala de laboratorio, no reproduce un reformador de metano ni emplea biogás, sino que emula la etapa de captura y monitoreo del CO₂ asociada al hidrógeno esmeralda, utilizando una fuente de gas sustituta como se describe más adelante.

Diseño del sensor de resonancia de Helmholtz. El sensor para medir la concentración de CO₂ fue diseñado y construido en la primera etapa del proceso. El resonador se formó en CAD como una cavidad cilíndrica con un cuello tubular (resonador tipo sifón), cuyo volumen y área de sección transversal determinan su

frecuencia de resonancia, como: $f_0 = \frac{c}{2\pi} \sqrt{\frac{A}{V \cdot L_{eq}}}$

Donde c es la velocidad del sonido en el gas, A el área de la sección transversal del cuello, V el volumen de la cavidad y L_{eq} la longitud equivalente del cuello. Para el resonador construido se emplearon un radio r= 6 mm y una longitud L=35 mm, de donde se obtuvieron $A = \pi r^2 = 1.13 \times 10^{-4} \text{ m}^2$, $V = \pi r^2 L = 3.958 \times 10^{-6} \text{ m}^3 (\approx 3958.4 \text{ mm}^3)$ y $L_{eq} = 35 \times 10^{-3} \text{ m}$. Es un hecho que la velocidad del sonido depende de la masa molar y la densidad de la mezcla de gases, y a medida que aumenta la fracción de CO₂, el gas se vuelve más denso, c disminuye, y por lo tanto la frecuencia de resonancia f₀ cambia. Para determinar la concentración de CO₂ con este cambio de frecuencia, la señal del sensor fue

adquirida desde una placa Arduino Due (puerto COM5) conectada a una computadora y procesada y graficada en Python.



Figura 1. Modelo CAD del resonador de Helmholtz (cavidad cilíndrica con cuello tubular).

Validación del modelo del resonador. A partir de la frecuencia de resonancia medida en cada prueba y los parámetros geométricos del resonador, se determinó la velocidad del sonido del gas $v = f_0 \cdot 2\pi \sqrt{\frac{V \cdot L_{eq}}{A}}$ para validar nuestro modelo. Los valores encontrados son 55.86 m/s y 57.71 m/s, que se compararon con las velocidades teóricas de 268 m/s para el CO₂ y 343 m/s para el aire. El error relativo de la relación entre la velocidad del sonido (teórica = 1.279 frente a la experimental = 1.03311) es de 19.2 %. Esta diferencia es coherente con el carácter de prueba de concepto del dispositivo y señala el margen de mejora del modelo geométrico empleado.

Como $2\pi\sqrt{(V \cdot L_{eq}/A)}$ depende únicamente de la geometría del resonador, igual en ambas pruebas, ese factor se cancela al comparar las dos configuraciones por cociente. Por eso se validó con $v = f_0\sqrt{(V \cdot L_{eq}/A)}$ auxiliar (proporcional a $c/2\pi$, $v = c/2\pi$) cuya

razón v_{aire}/v_{CO_2} es exactamente igual que c_{aire}/c_{CO_2} que compara frente a relación teórica pero no se ve afectado el resultado de comparación.

El desarrollo de los cálculos es:

$$35 \text{ mm} \quad V = \pi(6 \text{ mm})^2 (35 \text{ mm})$$

$$r = 6 \text{ mm}$$

$$V = 3958.4 \text{ mm}^3$$

$$3958.4 \text{ mm}^3 \times \frac{1 \text{ m}^3}{10^9 \text{ mm}^3}$$

$$= 3.958 \times 10^3 \times 10^{-9} \text{ m}^3$$

$$= 3.958 \times 10^{-6} \text{ m}^3$$

$$A = \pi r^2$$

$$A = \pi(6 \text{ mm})^2 \times \frac{1 \text{ m}^2}{10^6 \text{ mm}^2}$$

$$A = 1.13 \times 10^{-4} \text{ m}^2$$

$$f = 1595.6 \text{ Hz} \quad \text{Burbujeo}$$

$$f = 1648.4 \text{ Hz} \quad \text{Sin burbujeo}$$

$$5.90 \text{ mm}$$

$$5.40 \text{ mm}$$

$$Leq = 35 \times 10^{-3} \text{ m}$$

$$A = 1.13 \times 10^{-4} \text{ m}^2 \quad [=] \sqrt{\frac{\text{m}^2}{\text{m}^3 \cdot \text{m}}} = \sqrt{\frac{\text{m}^2}{\text{m}^4}} = \frac{1}{\text{m}}$$

$$V = 3.958 \times 10^{-6} \text{ m}^3$$

$$v = \frac{f}{\sqrt{\frac{A}{V_0 Leq}}} = f \sqrt{\frac{V_0 Leq}{A}}$$

$$v_{CO_2} = 55.86 \frac{\text{m}}{\text{s}}$$

$$v_{aire} = 57.71 \frac{\text{m}}{\text{s}}$$

Velocidades teóricas

$$v_{CO_2} = 268 \frac{\text{m}}{\text{s}}$$

$$v_{aire} = 343 \frac{\text{m}}{\text{s}}$$

$$f_E = 1.03311$$

$$\%E = \frac{|(1.279 - 1.03311)|}{1.279} \times 100$$

$$f_T = 1.279$$

$$\%E = 19.2\%$$

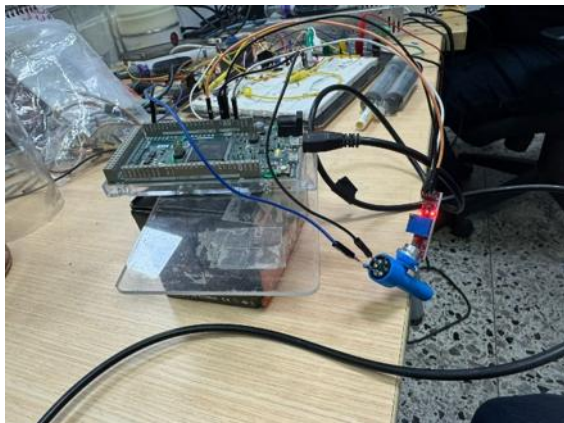
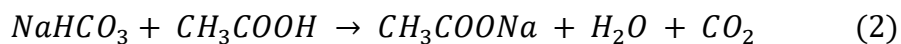


Figura 2. Montaje del sensor de resonancia de Helmholtz con la placa Arduino Due.

Diseño y construcción del reactor de columna de burbujeo. Una vez disponible el sistema de monitoreo, se construyó el reactor de columna de burbujeo destinado a la captura. El reactor contenía una solución de hidróxido de sodio (NaOH) como medio absorbente; al ingresar el gas por la parte inferior se formaban burbujas que aumentaban el área de contacto entre el CO₂ y la solución, favoreciendo la transferencia de masa y la absorción química del gas para formar carbonato de sodio. La ecuación de esta reacción es la misma presentada en la ecuación (1).

Generación de CO₂. El dióxido de carbono se generó mediante la reacción entre bicarbonato de sodio (NaHCO₃) y ácido acético (CH₃COOH) en un recipiente con tapa negra cerrado.



El gas generado se condujo por una manguera desde el recipiente de reacción hacia el reactor de columna de burbujeo.



Figura 3. Sistema de generación de CO_2 mediante la reacción de bicarbonato de sodio con ácido acético conectado al reactor de columna de burbujeo.

Integración y diseño experimental. Con los tres elementos en funcionamiento se realizaron dos pruebas bajo condiciones idénticas, pero se utilizaron diferentes posiciones de la manguera de entrada para determinar el efecto del burbujeo.

Prueba con burbujeo. En la prueba con burbujeo, la manguera se mantuvo conectada al fondo del reactor, y el CO_2 fluiría a través de la solución de NaOH en burbujas, para asegurar que el gas y el absorbente pasen a través de la solución de NaOH .

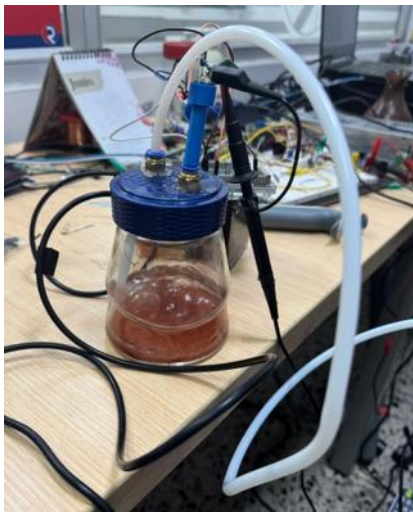


Figura 4. Montaje de la prueba con burbujeo: la manguera de entrada se conecta hasta el fondo del reactor.

Prueba sin burbujeo. En la prueba sin burbujeo, la manguera no estaba conectada al interior del reactor para evitar la formación de burbujas y limitar el contacto del CO₂ con la solución.



Figura 5. Montaje de la prueba sin burbujeo: la manguera de entrada no entra en contacto con la solución.

Adquisición y procesamiento de datos. Cada prueba generó una serie temporal de f_0 que se exportó en formato CSV y se procesó en Python para su limpieza, representación gráfica (f_0 frente al tiempo) y cálculos estadísticos. La prueba con burbujeo se registró durante 4.5 min con 10 muestras y la prueba sin burbujeo durante 5.0 min con 11 muestras.

Resultados y Discusión

Análisis estadístico.

Para cada serie, se calcularon estadísticas descriptivas para la media, desviación estándar, valores mínimos y máximos, y coeficiente de variación (ver Tabla 1).

Tabla 1. Estadísticos descriptivos de la frecuencia de resonancia f_0 en cada configuración.

Parámetro	Con burbujeo	Sin burbujeo
Muestras	10	11
Duración	4.5 min	5.0 min
f_0 media (Hz)	1595.6	1648.4
Desviación estándar (Hz)	167.5	87.0
Mínimo	1182	1440
Máximo (Hz)	1774	1717
Coeficiente de variación	10.5 %	5.3 %

Ambas pruebas pueden registrar la frecuencia de resonancia f_0 y la concentración de CO_2 en el sistema. La razón física de esta lectura es que cuando se incrementa la concentración de CO_2 , la mezcla de gases se vuelve más densa y la velocidad del sonido disminuye, por lo tanto, f_0 disminuye. En ambos casos, el sensor comenzó desde valores bajos de f_0 que corresponden a la alta concentración de CO_2 que se liberó al inicio de la reacción entre bicarbonato y ácido acético y aumentó gradualmente hasta 1700 Hz, lo cual correspondió a la disminución de CO_2 cuando el gas se difunde o es absorbido por la solución de NaOH. Los valores a continuación (media, desviación estándar, mínimo,

máximo, coeficiente de variación) fueron calculados a partir de la serie temporal de cada prueba registrada en las Tablas 2 y 3 y resumidos en la Tabla 1.

Prueba con burbujeo.

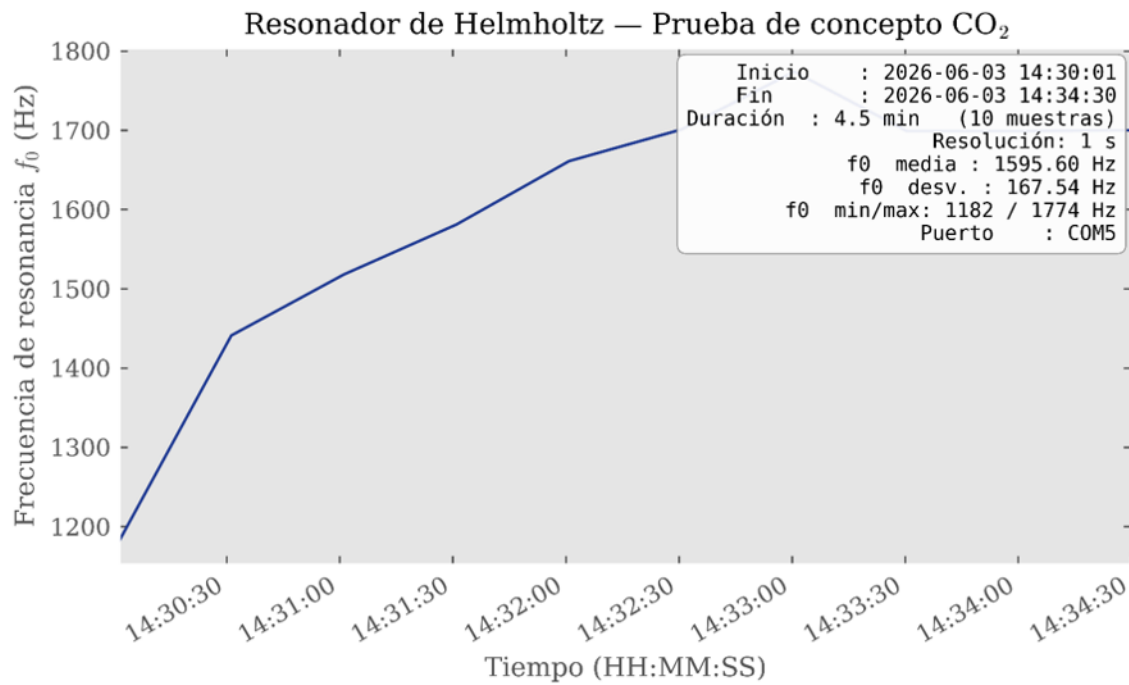


Figura 6. Evolución temporal de la frecuencia de resonancia f_0 en la prueba con burbujeo.

Tabla 2. Datos registrados en la prueba con burbujeo.

# Sensor	Resonador de Helmholtz (PoC)
# Puerto	COM5
# Inicio	2026-06-03 14:30:01
# Fin	2026-06-03 14:34:30
# Muestras	10
Fecha y Hora	f0 (Hz)
3/06/2026 14:30	1182
3/06/2026 14:30	1441
3/06/2026 14:31	1518
3/06/2026 14:31	1581
3/06/2026 14:32	1661
3/06/2026 14:32	1701
3/06/2026 14:33	1774
3/06/2026 14:33	1699
3/06/2026 14:34	1699
3/06/2026 14:34	1700

En la prueba de burbujeo (Figura 6 y Tabla 2), f_0 comenzó en 1182 Hz, el valor más bajo de toda la serie, y este es el pico transitorio de la mayor concentración de CO_2 en los segundos iniciales de la reacción. Luego, la frecuencia aumentó hasta estabilizarse alrededor de 1700 Hz, y se produjo un sobreimpulso transitorio de 1774 Hz a las 14:33. A partir de los diez puntos de datos en la Tabla 2, se pueden obtener las estadísticas en la Tabla 1 de la siguiente manera, la media (1595.6 Hz) es el promedio de las diez muestras; la desviación estándar (167.5 Hz) indica cuán alejados están los valores de ese promedio;

el rango (592 Hz) es la diferencia entre el máximo (1774 Hz) y el mínimo (1182 Hz). El coeficiente de variación (10.5%) se obtiene dividiendo la desviación estándar por la media (167.5 / 1595.6). Este gran coeficiente y el amplio rango pueden interpretarse en términos de la respuesta dinámica de la señal donde es muy alta y oscila en la Figura 6.

Prueba sin burbujeo.

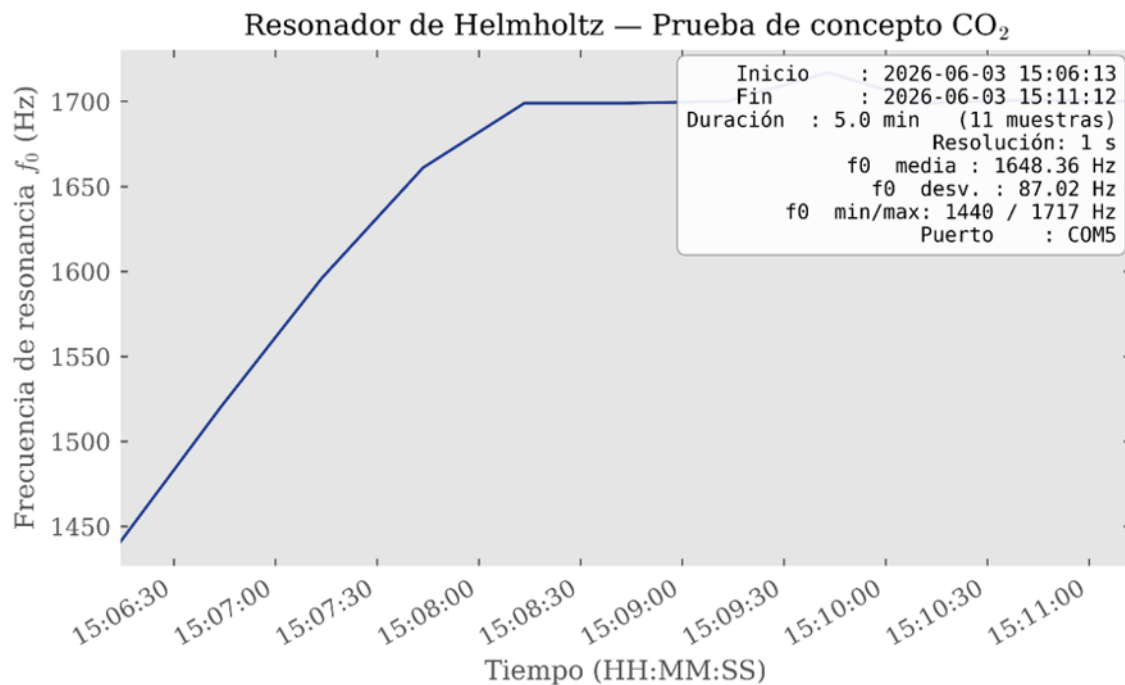


Figura 7. Evolución temporal de la frecuencia de resonancia f_0 en la prueba sin burbujeo.

Tabla 3. Datos registrados en la prueba sin burbujeo.

# Sensor	Resonador de Helmholtz (PoC)
# Puerto	COM5
# Inicio	2026-06-03 15:06:13
# Fin	2026-06-03 15:11:12
# Muestras	11
Fecha y Hora	f0 (Hz)
3/06/2026 15:06	1440
3/06/2026 15:06	1520
3/06/2026 15:07	1596
3/06/2026 15:07	1661
3/06/2026 15:08	1699
3/06/2026 15:08	1699
3/06/2026 15:09	1700
3/06/2026 15:09	1717
3/06/2026 15:10	1699
3/06/2026 15:10	1701
3/06/2026 15:11	1700

En la prueba sin burbujeo (Figura 7 y Tabla 3), f_0 comenzó en 1440 Hz, un punto de inicio más alto que en la prueba anterior, y mejoró con el tiempo hasta alcanzar el rango estable (1700 Hz) en menos de un minuto y permaneció allí hasta el final. De los once puntos de datos en la Tabla 3, podemos obtener las estadísticas en la Tabla 1 como antes: la media (1648.4 Hz) es el promedio de las once muestras; la desviación estándar (87.0 Hz) es casi la mitad de lo que sería si se usara burbujeo; el rango (277 Hz) es la

diferencia entre el máximo (1717 Hz) y el mínimo (1440 Hz); y el coeficiente de variación (5.3%) resulta de dividir 87.0 por 1648.4. Estos valores son claramente más bajos que los de la prueba con burbujeo, y también muestran un comportamiento mucho más estable (ver la curva más suave en la Figura 7).

Comparación entre pruebas.

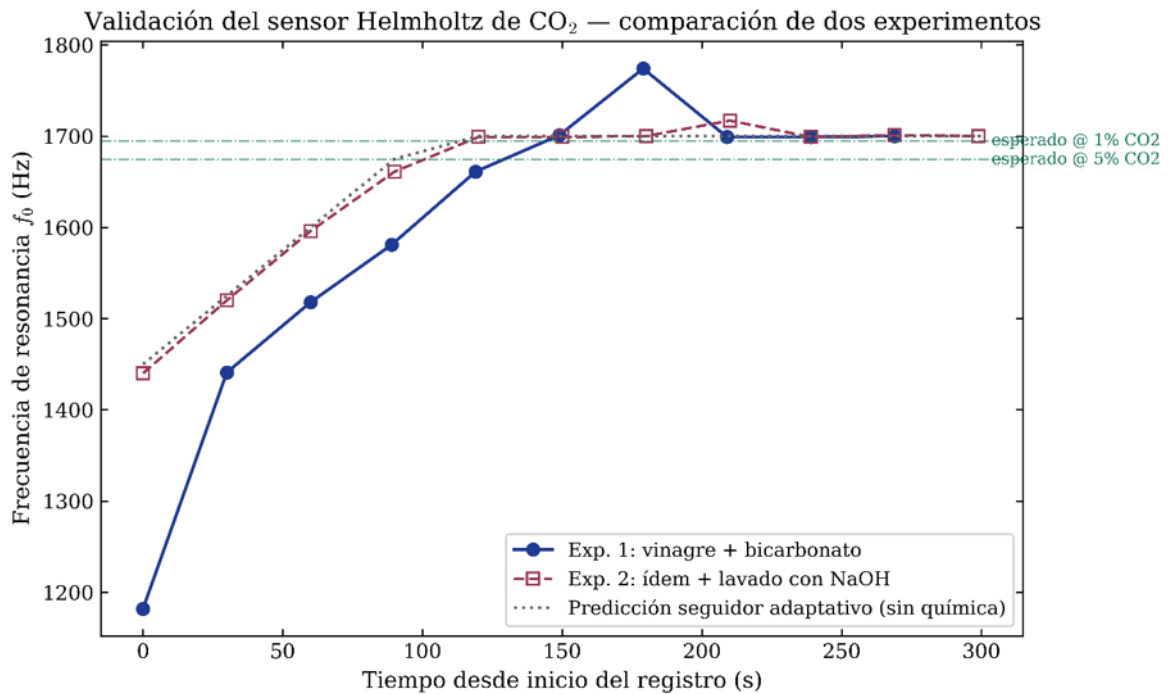


Figura 8. Comparación de la frecuencia de resonancia f_0 en ambas pruebas frente a las líneas esperadas para 1 % y 5 % de CO₂.

La comparación entre las dos configuraciones muestra que cuando hay burbujeo el gas se acerca más a la solución de NaOH y así una respuesta dinámica con mayor dispersión resulta; cuando no hay burbujeo el sistema se estabiliza más rápidamente y menos variabilidad tiene. En general los resultados también muestran que el sensor de resonancia Helmholtz siempre siguió en tiempo real la evolución del CO₂ en las dos

configuraciones estudiadas. Por lo tanto, estos coeficientes de variación (10.5 % frente 5.3 %), tasa inicial de cambio (8.9 Hz/s frente a 2.7 Hz/s) sobreimpulso (4.4 % frente a 1.0 %) confirman de manera consistente que el burbujeo produce una respuesta más rápida pero más ruidosa (mayor dispersión) mientras que prueba sin burbujeo da lectura más estable y predecible.

Dinámica de respuesta del sensor. De los datos ya registrados en las Tablas 2 y 3 hemos obtenido dos resultados adicionales sobre la dinámica de respuesta del sensor, tasa inicial de cambio f_0 y magnitud del sobreimpulso (que también están incluidos en el análisis estadístico de la Tabla 1). Como el registro no guarda los segundos de cada lectura individual, se usó el intervalo promedio de muestreo (duración total $\div$ número de intervalos entre muestras): con 10 muestras (9 intervalos) y 269 s de duración, $269 \text{ s} \div 9 \approx 29.9 \text{ s}$ con burbujeo; con 11 muestras (10 intervalos) y 299 s de duración, $299 \text{ s} \div 10 \approx 29.9 \text{ s}$ sin burbujeo (duraciones y N° de muestras de la Tabla 2 y la Tabla 3). Con ese intervalo, la tasa inicial se calculó como la diferencia entre las dos primeras lecturas dividida entre el intervalo: $(1441 - 1182) \text{ Hz} \div 29 \text{ s} \approx 8.9 \text{ Hz/s}$ con burbujeo, y $(1520 - 1440) \text{ Hz} \div 30 \text{ s} \approx 2.7 \text{ Hz/s}$ sin burbujeo; es decir, la respuesta inicial del sensor fue casi tres veces más rápida cuando se forzó el contacto gas-líquido mediante burbujeo (lo que está consistente con mayor área transferencia masa); La prueba con burbujeo además tenía sobreimpulso de f_0 hasta 1774 Hz hacia los 3 min de ensayo (unos 74-75 Hz por encima del nivel estable (1699-1700 Hz)) lo que da una desviación transitoria del 4.4 %; en la prueba sin burbujeo solo hubo una desviación máxima respecto al nivel estable de 17-18 Hz (1.0 %) pero no hay sobreimpulso apreciable (esto refuerza lo visto en

coeficiente variación (10.5 % frente a 5.3 %, Tabla 1). Tiempo de estabilización, sin burbujeo alcanzó su nivel estable (1699 Hz) en unos 2 min y estuvo prácticamente constante durante los 3 minutos restantes del ensayo, con burbujeo alcanzó un valor cercano (1701 Hz) en tiempo similar (2.5 min), pero tardó cerca de 1 min más en estabilizarse definitivamente debido al sobreimpulso descrito.

Conclusiones.

El prototipo desarrollado en este trabajo permitió la monitorización en tiempo real y la captura de CO_2 simultáneamente en la etapa de reformado de metano que da lugar al hidrógeno esmeralda. El sensor basado en resonancia de Helmholtz combinado con la placa Arduino Due y el algoritmo de Python fue capaz de seguir continuamente la evolución del CO_2 en las dos configuraciones consideradas, y el reactor de columna de burbujeo con hidróxido de sodio fue capaz de capturar el CO_2 generado, reteniéndolo en forma de carbonato de sodio. La comparación de los dos escenarios experimentales con y sin burbujeo muestra que el contacto forzado entre el gas y la solución alcalina da una respuesta más fuerte con mayor dispersión (10.5% de coeficiente de variación), y si no se aplica burbujeo, la reacción es más lenta pero más estable y menos ruidosa (5.3% de coeficiente de variación). Esto podría significar que cuando el gas tiene que ser burbujeado, transferirá más masa, pero a costa de ruido. El error relativo del 19.2% del modelo de resonador con la velocidad del sonido teórica proporciona la prueba de concepto de la investigación e indica que hay mucho margen de mejora en el modelo geométrico. Los resultados sugieren que un sistema fácil de usar y de bajo costo es factible para validar el proceso de monitorización y captura de CO_2 del hidrógeno esmeralda y sentar las bases para futuros trabajos para mejorar la precisión del sensor y el rango de concentraciones, y escalar la configuración a una planta real.

Referencias

- Alahmari, S. a.-W. (2019). Diode laser photoacoustic spectroscopy of CO₂, H₂S and O₂ in a differential Helmholtz resonator for trace gas analysis in the biosciences and petrochemistry: Alahmari S. et al. *Analytical and bioanalytical chemistry*, 3777--3787.
- Ameri, A. a. (2025). Comparative study of CO₂ capture efficiency using sodium hydroxide and monoethanolamine solutions in a spray column. *Atmospheric Environment: X*, 100383.
- Anand, C. a. (2025). Green hydrogen for a sustainable future: A review of production methods, innovations, and applications. *International Journal of Hydrogen Energy*, 319--341.
- Berndt, D. a. (2022). A Robust Miniaturized Gas Sensor for H₂ and CO₂ Detection Based on the 3 ω Method. *Sensors*, 485.
- Blackstock, J. M. (2019). Monitoring atmospheric, soil, and dissolved CO₂ using a low-cost, Arduino monitoring platform (CO₂-LAMP): Theory, fabrication, and operation. *Frontiers in Earth Science*, 313.
- Cao, Y. a. (2024). Breakthrough applications of porous organic materials for membrane-based CO₂ separation: a review. *Frontiers in Chemistry*, 1381898.
- Chan, R.-Y. a.-Z.-C.-C.-W. (2025). Experimental study of carbon dioxide capture and mineral carbonation using sodium hydroxide solution. *Journal of Ecological Engineering*.
- Cho, H. H. (2022). Environmental impact assessment of hydrogen production via steam methane reforming based on emissions data. *Energy Reports*, 13585--13595.
- Correia, P. a. (2023). Understanding the role of mono and ternary alkali metal salts on CO₂ uptake of MgO sorbents. *Materials*, 7539.
- Fleischer, C. a. (1996). Detailed modeling of the chemisorption of CO₂ into NaOH in a bubble column. *Chemical engineering science*, 1715--1724.
- Grinias, J. P. (2016). An inexpensive, open-source USB Arduino data acquisition device for chemical instrumentation. *Journal of Chemical Education*, 1316.
- Gul, A. a. (2023). Carbon Dioxide Absorption Using Different Solvents (MEA, NaOH, KOH and Mg (OH) ₂) in Bubble Column Reactor. *Bitlis Eren $\{^{\text{U}}\}$ niversitesi Fen Bilimleri Dergisi*, 418--427.
- Hamed, H. a.-C. (2025). Ex situ carbon mineralization for CO₂ capture using industrial alkaline wastes—optimization and future prospects: a review. *Clean Technologies*, 44.
- Jumah, A. b. (2024). A comprehensive review of production, applications, and the path to a sustainable energy future with hydrogen. *RSC advances*, 26400--26423.
- Keeratrirawee, K. a. (2022). Determination of binary gas mixtures by measuring the resonance frequency in a piezoelectric tube. *Sensors*, 1691.
- Kumar, S. a. (2014). A comparative study of CO₂ sorption properties for different oxides. *Materials for Renewable and Sustainable Energy*, 30.
- Lopez, K. P.-M. (2026). El agua como membrana de separación de gases. *Nature Communications*.

- Michelson, J. a.-A. (2024). Modeling of a bubble column for CO₂ removal by absorption with NaOH. *Chemical Engineering Communications*, 571--581.
- Nagireddi, S. a. (2024). Carbon dioxide capture, utilization, and sequestration: current status, challenges, and future prospects for global decarbonization. *ACS Engineering Au*, 22--48.
- Nava, A. a. (2024). Carbon-negative “emerald hydrogen” from electrified steam methane reforming of biogas: System integration and optimization. *International Journal of Hydrogen Energy*, 1237--1255.
- Navas-Anguila, Z. a.-G. (2021). Revisiting the role of steam methane reforming with CO₂ capture and storage for long-term hydrogen production. *Science of the total Environment*, 145432.
- Pino, H. a.-{. (2019). Measuring CO₂ with an Arduino: creating a low-cost, pocket-sized device with flexible applications that yields benefits for students and schools. *Journal of Chemical Education*.
- Reivanth, K. a. (2024). A Low-Cost Arduino and Python Based Gas Sensing Setup: Bridging Theory and Practice in Educational Environments. *Journal Of Chemical Education*, 5361--5368.
- Sa'adiyah, D. S. (2021). Effects of chemical absorption on mass transfer from single carbon dioxide bubbles in aqueous sodium hydroxide solution in a vertical pipe. *Chemical engineering science*, 116852.
- Suchenek, M. a. (2018). Measuring sound speed in gas mixtures using a photoacoustic generator. *International Journal of Thermophysics*, 11.
- Sun, N. a. (2015). Solid adsorbents for low-temperature CO₂ capture with low-energy penalties leading to more effective integrated solutions for power generation and industrial processes. *Frontiers in Energy Research*, 9.
- Tiefenthaler, J. a. (2022). Experimental investigation of a continuous reactor for CO₂ capture and CaCO₃ precipitation. *Frontiers in Chemical Engineering*, 830284.
- van der Zwaan, B. a. (2025). Electricity-and hydrogen-driven energy system sector-coupling in net-zero CO₂ emission pathways. *Nature Communications*, 1368.
- Xu, X. a. (2021). High-efficiency CO₂ separation using hybrid LDH-polymer membranes. *Nature Communications*, 3069.
- Younus, H. A.-J. (2025). Green hydrogen production and deployment: opportunities and challenges. *Discover Electrochemistry*, 32.
- Zaky, Z. A. (2025). Novel Helmholtz resonators parallel to lateral closed resonators for sensing applications. *Scientific Reports*, 37746.

Anexos

Anexo A. Desarrollo numérico de la validación del modelo del resonador. En este anexo se muestra paso a paso el desarrollo numérico que llevó al error relativo del 19.2 % reportado en la sección “Validación del modelo del resonador”. la velocidad del sonido solo con la frecuencia medida y las dimensiones ya conocidas del dispositivo. Se usó la ecuación del resonador de Helmholtz ($f_0 = (v / 2\pi) \cdot \sqrt{A / (V \cdot L_{eq})}$) porque describe el comportamiento acústico de una cavidad cilíndrica con cuello tubular (es decir, geometría del resonador construido en Figura 1) y permite estimar la velocidad del sonido del gas sólo con la frecuencia medida y las dimensiones ya conocidas del dispositivo: despejando v y usando las frecuencias medias de cada prueba (1595.6 Hz y 1648.4 Hz Tabla 1) con las dimensiones del resonador ($r = 6 \text{ mm}$ $L = L_{eq} = 35 \text{ mm}$), se tuvieron $v_{CO_2} = 55.86 \text{ m/s}$ y $v_{aire} = 57.71 \text{ m/s}$ que se compararon por cociente ($v_{aire}/v_{CO_2} = 1.03311$) frente a la razón teórica ($343/268 = 1.279$) pues así cancelan los factores geométricos comunes y aislamos la velocidad solo para cada caso concreto. Cabe señalar que v_{CO_2} y v_{aire} son la cantidad auxiliar $v = c/2\pi$ (no la velocidad absoluta del sonido que es 2π veces mayor: 350.9 m/s y $\approx 362.5 \text{ m/s}$). Este 2π es un factor geométrico común en ambas pruebas, por lo que no afecta la relación v_{aire}/v_{CO_2} ni el error relativo del 19.2%.

Anexo B. Código del microcontrolador (Arduino Due). Sketch

Helmholtz_CO2.ino desarrollado por el profesor Juan Esteban Velez Alvarez. (tutor de este trabajo de grado) para generar la señal de excitación del resonador mediante DAC,

adquirir la señal del piezoeléctrico, ubicar f_0 con algoritmo de Goertzel y transmitirla por serie puerto.

```

/*
  Resonador de Helmholtz – DAC con escritura directa al registro
  =====
  Cambios respecto a la versión anterior:
  1) setup() inicializa el DAC0 con un analogWrite "ficticio"
    (deja al núcleo Arduino configurar pmc / channel / mode).
  2) La ISR YA NO llama a analogWrite – ahora escribe
    directamente DAC->DACC_CDR. Es ~10x más rápido y
    elimina el "while (TXRDY==0)" bloqueante dentro de la ISR.
  3) Sigue produciendo escalera; añadir filtro RC al pin DAC0:
      DAC0 -- 1.0 kOhm -- piezo(+)
                |
                100 nF
                |
      GND -----+--- piezo(-)
*/

#include <Arduino.h>

// ----- Parámetros -----
static const uint32_t FS_HZ      = 20000;
static const float    F_CENTER_0 = 1200.0f;
static const float    SWEEP_HALF = 250.0f;
static const float    SWEEP_STEP = 2.0f;
static const uint16_t N_SAMPLES  = 1000;
static const uint16_t DAC_AMP     = 600;
static const uint16_t DAC_MID     = 2048;
static const uint8_t  ADC_PIN     = A0;

// ----- DDS -----
volatile uint32_t dds_phase = 0;
volatile uint32_t dds_inc   = 0;
static int16_t sineTable[1024];

void buildSineTable() {
  for (int i = 0; i < 1024; ++i) {
    sineTable[i] = (int16_t)lroundf(2047.0f * sinf(2.0f * PI * i / 1024.0f));
  }
}

inline void setFrequency(float f) {
  dds_inc = (uint32_t)((double)f * 4294967296.0 / (double)FS_HZ);
}

// ===== ISR rápida: escritura directa al CDR =====
void TC3_Handler() {
  TC_GetStatus(TC1, 0);
  dds_phase += dds_inc;
  uint16_t idx = dds_phase >> 22;
  int16_t s    = sineTable[idx];
  int32_t v    = DAC_MID + ((int32_t)s * DAC_AMP) / 2047;
  if (v < 0)    v = 0;
  if (v > 4095) v = 4095;

  // Escritura directa al registro de datos de conversión del DAC.
  // No hay polling de TXRDY: si por alguna razón el DAC todavía
  // no está listo, se pierde UNA muestra y la siguiente ISR sigue.
  // En la práctica, a 20 kHz contra un DAC que puede a 500 kHz,
  // esto nunca ocurre.
  DAC->DACC_CDR = (uint32_t)v;
}

```

```

}

void startSampleTimer(uint32_t fs) {
    pmc_set_writeprotect(false);
    pmc_enable_periph_clk(ID_TC3);
    TC_Configure(TC1, 0,
        TC_CMR_WAVE | TC_CMR_WAVSEL_UP_RC |
        TC_CMR_TCCLKS_TIMER_CLOCK1);
    uint32_t rc = (uint32_t)(4200000UL / fs);
    TC_SetRA(TC1, 0, rc / 2);
    TC_SetRC(TC1, 0, rc);
    TC_Start(TC1, 0);
    TC1->TC_CHANNEL[0].TC_IER = TC_IER_CPCS;
    TC1->TC_CHANNEL[0].TC_IDR = ~TC_IER_CPCS;
    NVIC_EnableIRQ(TC3_IRQn);
}

// ----- DAC: dejar que analogWrite inicialice, luego liberar -----
void setupDAC() {
    analogWriteResolution(12);
    // Un analogWrite "ficticio" hace todo el setup del periférico:
    // habilita PMC del DAC, selecciona el canal 0 (=DAC0), pone
    // el modo, etc. Después la ISR escribe DAC_CDR directamente.
    analogWrite(DAC0, DAC_MID);
}

void setupADC() {
    analogReadResolution(12);
    for (int i = 0; i < 8; ++i) (void)analogRead(ADC_PIN);
}

// ----- Goertzel -----
float goertzelMagSq(float targetHz) {
    float k = 0.5f + ((float)N_SAMPLES * targetHz) / (float)FS_HZ;
    float w = (2.0f * PI / (float)N_SAMPLES) * (int)k;
    float coeff = 2.0f * cosf(w);
    uint32_t periodo_us = 1000000UL / FS_HZ;

    // 1ra pasada: estimar DC
    uint32_t t0 = micros();
    float dc = 0.0f;
    for (uint16_t n = 0; n < N_SAMPLES; ++n) {
        while ((uint32_t)(micros() - t0) < (uint32_t)n * periodo_us) { }
        dc += analogRead(ADC_PIN);
    }
    dc /= (float)N_SAMPLES;

    // 2da pasada: Goertzel sobre (x - dc)
    float s_prev = 0.0f, s_prev2 = 0.0f;
    t0 = micros();
    for (uint16_t n = 0; n < N_SAMPLES; ++n) {
        while ((uint32_t)(micros() - t0) < (uint32_t)n * periodo_us) { }
        float x = (float)analogRead(ADC_PIN) - dc;
        float s = x + coeff * s_prev - s_prev2;
        s_prev2 = s_prev;
        s_prev = s;
    }
    return s_prev2 * s_prev2 + s_prev * s_prev - coeff * s_prev * s_prev2;
}

// ----- Búsqueda de pico -----
float f_center = F_CENTER_0;

float findPeakOnce() {
    float bestF = f_center;
    float bestMag = -1.0f;
}

```

```

    for (float f = f_center - SWEEP_HALF; f <= f_center + SWEEP_HALF; f += SWEEP_STEP) {
        setFrequency(f);
        delay(20);
        float mag = goertzelMagSq(f);
        if (mag > bestMag) { bestMag = mag; bestF = f; }
    }
    return bestF;
}

// ----- Setup / Loop -----
void setup() {
    Serial.begin(115200);
    unsigned long t = millis();
    while (!Serial && (millis() - t) < 3000) { }

    Serial.println(F("# Helmholtz C02 PoC v1.1 (Arduino Due)"));
    Serial.println(F("# ISR usa escritura directa DACC_CDR"));

    buildSineTable();
    setupDAC();
    setupADC();
    setFrequency(F_CENTER_0);
    startSampleTimer(FS_HZ);

    Serial.print(F("# Fs="));
    Serial.print(FS_HZ);
    Serial.println(F(" Hz Listo"));
    delay(500);
}

void loop() {
    float f0 = findPeakOnce();
    Serial.print(F("# pico f0="));
    Serial.print(f0, 1);
    Serial.println(F(" Hz"));
    Serial.println((int)lroundf(f0));
    f_center = 0.7f * f_center + 0.3f * f0;
}

```

Anexo C. Código de monitoreo y graficación (Python). Script

monitor_helmholtz.py escrito por el profesor Juan Esteban Velez Alvarez. (tutor de este trabajo de grado) encargado de detectar puerto Arduino, graficar en tiempo real f_0 de resonancia, grabar cada lectura en archivo CSV y exportar figuras finales PNG (600 dpi) y PDF.

```

"""
Monitor del resonador de Helmholtz – Prueba de concepto C02
=====
Compañera del sketch Helmholtz_C02.ino (Arduino Due).

- Detección automática del puerto serie.
- Eje X: tiempo real ; Eje Y: frecuencia de resonancia  $f_0$  (Hz).
- Resolución temporal seleccionable: 1 s / 10 s / 1 min.
- Registro continuo en CSV.

```

```

- Exportación a PNG (600 dpi) y PDF: cada botón construye una
  figura NUEVA, limpia, con tipografía y márgenes de paper, e
  incrusta los metadatos del experimento – no es una captura
  de la ventana del GUI.
"""

import serial
import serial.tools.list_ports
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.animation import FuncAnimation
from matplotlib.widgets import RadioButtons, Button
from collections import deque
from datetime import datetime
import csv
import os
import sys
import statistics

# --- CONFIGURACIÓN ---
CARPETA_BASE = os.path.join(os.path.expanduser("~"), "Desktop", "clases", "co2")
ARCHIVO_LOG = os.path.join(CARPETA_BASE, "helmholtz_f0.csv")
BAUDIOS = 115200
MAX_PUNTOS = 600 # ventana deslizable

# Estado mutable accesible dentro de los callbacks
estado = {
    'resolucion_s' : 1,
    'ultimo_registro': None,
    'puerto' : None,
    'primer_dato' : None,
    'ultima_lectura' : None,
    'mensaje_serial' : '',
    'tiempo_conexion': None,
    'ultimo_f0' : None,
}

tiempos = deque(maxlen=MAX_PUNTOS)
valores = deque(maxlen=MAX_PUNTOS)

os.makedirs(CARPETA_BASE, exist_ok=True)

# --- ESCANEAO AUTOMÁTICO DEL PUERTO ARDUINO ---
def buscar_arduino():
    palabras_clave = (
        'arduino', 'ch340', 'ch341', 'usb-serial', 'usb serial',
        'cp210', 'ftdi', 'wchusbserial', 'usbmodem', 'usbserial',
    )
    puertos = list(serial.tools.list_ports.comports())
    for p in puertos:
        txt = ' '.join(filter(None, [p.description, p.manufacturer, p.device])).lower()
        if any(k in txt for k in palabras_clave):
            return p.device
    for p in puertos:
        try:
            s = serial.Serial(p.device, BAUDIOS, timeout=0.5); s.close()
            return p.device
        except (OSError, serial.SerialException):
            continue
    return None

PUERTO = buscar_arduino()
if not PUERTO:
    print("Error: no se detectó ningún Arduino conectado.")

```

```

sys.exit(1)

try:
    arduino = serial.Serial(PUERTO, BAUDIOS, timeout=1)
    estado['puerto'] = PUERTO
    estado['tiempo_conexion'] = datetime.now()
    print(f"Conectado en {PUERTO}")
    print(f"Guardando en : {ARCHIVO_LOG}")
except serial.SerialException as e:
    print(f"Error al abrir {PUERTO}: {e}")
    sys.exit(1)

# --- CSV ---
if not os.path.exists(ARCHIVO_LOG):
    with open(ARCHIVO_LOG, 'w', newline='', encoding='utf-8') as f:
        f.write("sep=\n")
        csv.writer(f, delimiter=';').writerow(
            ["Fecha y Hora", "f0 (Hz)", "Resolucion (s)"]
        )

# --- FIGURA GUI ---
plt.style.use('ggplot')
fig, ax = plt.subplots(figsize=(10.5, 5.8))
plt.subplots_adjust(left=0.30, bottom=0.16, right=0.97, top=0.92)
try:
    fig.canvas.manager.set_window_title("Monitor Helmholtz CO2")
except Exception:
    pass

linea, = ax.plot([], [], linewidth=1.6, color='#1f3a93')
ax.set_xlabel("Tiempo")
ax.set_ylabel(r"Frecuencia de resonancia $f_0$ (Hz)")
ax.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))

# Radio buttons: resolución temporal
ax_radio = plt.axes([0.02, 0.55, 0.16, 0.27], facecolor='lightgoldenrodyellow')
radio = RadioButtons(ax_radio, ('1 s', '10 s', '1 min'), active=0)
ax_radio.set_title("Resolución temporal", fontsize=10, pad=6)

def cambiar_resolucion(label):
    mapa = {'1 s': 1, '10 s': 10, '1 min': 60}
    estado['resolucion_s'] = mapa[label]
    estado['ultimo_registro'] = None
    tiempos.clear(); valores.clear()
    print(f"Resolución -> {label}")

radio.on_clicked(cambiar_resolucion)

# --- FIGURA DE PUBLICACIÓN (se construye al exportar) ---
def construir_figura_publicacion():
    """Crea una figura limpia, tipografía serif, con metadatos.

    Devuelve (fig, ruta_sugerida_sin_extension). El llamador escoge
    extensión y dpi.
    """
    if len(tiempos) < 2:
        return None, None

    # Estilo de paper local a esta figura, sin tocar el GUI
    pub_rc = {

```

```

        'font.family'      : 'serif',
        'mathtext.fontset': 'cm',
        'axes.labelsize'  : 11,
        'axes.titlesize'  : 12,
        'xtick.labelsize' : 10,
        'ytick.labelsize' : 10,
        'xtick.direction' : 'in',
        'ytick.direction' : 'in',
        'xtick.top'       : True,
        'ytick.right'     : True,
        'axes.linewidth'  : 0.8,
        'lines.linewidth' : 1.2,
        'axes.grid'       : False,
    }
    with plt.rc_context(pub_rc):
        fig_p, ax_p = plt.subplots(figsize=(6.8, 4.2))

        t_list = list(tiempos)
        v_list = list(valores)
        ax_p.plot(t_list, v_list, color='#1f3a93')

        ax_p.set_xlabel("Tiempo (HH:MM:SS)")
        ax_p.set_ylabel(r"Frecuencia de resonancia $f_0$ (Hz)")
        ax_p.set_title("Resonador de Helmholtz – Prueba de concepto CO$_2$")

        ax_p.xaxis.set_major_formatter(mdates.DateFormatter('%H:%M:%S'))
        ax_p.set_xlim(t_list[0], t_list[-1])

        v_min, v_max = min(v_list), max(v_list)
        margen      = max(2.0, 0.05 * (v_max - v_min))
        ax_p.set_ylim(v_min - margen, v_max + margen)

        # Estadísticas
        media      = statistics.fmean(v_list)
        desv       = statistics.pstdev(v_list) if len(v_list) > 1 else 0.0
        duracion   = (t_list[-1] - t_list[0]).total_seconds()

        meta = (
            f"Inicio      : {t_list[0].strftime('%Y-%m-%d %H:%M:%S')}\n"
            f"Fin         : {t_list[-1].strftime('%Y-%m-%d %H:%M:%S')}\n"
            f"Duración    : {duracion/60:.1f} min   ({len(v_list)} muestras)\n"
            f"Resolución: {estado['resolucion_s']} s\n"
            f"$f_0$ media : {media:.2f} Hz\n"
            f"$f_0$ desv. : {desv:.2f} Hz\n"
            f"$f_0$ min/max: {v_min:.0f} / {v_max:.0f} Hz\n"
            f"Puerto      : {estado['puerto']}"
        )
        ax_p.text(
            0.985, 0.97, meta,
            transform=ax_p.transAxes,
            ha='right', va='top',
            family='monospace', fontsize=8.5,
            bbox=dict(boxstyle='round,pad=0.45',
                    facecolor='white', edgecolor='#999999',
                    linewidth=0.6, alpha=0.92),
        )

        fig_p.autofmt_xdate()
        fig_p.tight_layout()

        base = f"helmholtz_f0_{datetime.now().strftime('%Y%m%d_%H%M%S')}"
        return fig_p, os.path.join(CARPETA_BASE, base)

# Botón: exportar PNG 600 dpi
ax_btn_png = plt.axes([0.02, 0.42, 0.16, 0.07])

```

```

btn_png = Button(ax_btn_png, 'Exportar PNG 600 dpi')

def exportar_png(_):
    fig_p, base = construir_figura_publicacion()
    if fig_p is None:
        print("Aún no hay suficientes datos para exportar.")
        return
    ruta = base + ".png"
    fig_p.savefig(ruta, dpi=600, bbox_inches='tight')
    plt.close(fig_p)
    print(f"PNG (600 dpi) guardado: {ruta}")

btn_png.on_clicked(exportar_png)

# Botón: exportar PDF vectorial
ax_btn_pdf = plt.axes([0.02, 0.32, 0.16, 0.07])
btn_pdf = Button(ax_btn_pdf, 'Exportar PDF vectorial')

def exportar_pdf(_):
    fig_p, base = construir_figura_publicacion()
    if fig_p is None:
        print("Aún no hay suficientes datos para exportar.")
        return
    ruta = base + ".pdf"
    fig_p.savefig(ruta, bbox_inches='tight')
    plt.close(fig_p)
    print(f"PDF guardado: {ruta}")

btn_pdf.on_clicked(exportar_pdf)

# Botón: exportar CSV (copia con encabezado completo)
ax_btn_csv = plt.axes([0.02, 0.22, 0.16, 0.07])
btn_csv = Button(ax_btn_csv, 'Exportar CSV anotado')

def exportar_csv(_):
    if len(tiempos) < 1:
        print("No hay datos para exportar.")
        return
    ruta = os.path.join(
        CARPETA_BASE,
        f"helmholtz_f0_{datetime.now().strftime('%Y-%m-%d_%H%M%S')}.csv"
    )
    with open(ruta, 'w', newline='', encoding='utf-8') as f:
        f.write("sep=;\n")
        f.write(f"# Sensor      ; Resonador de Helmholtz (PoC)\n")
        f.write(f"# Puerto      ; {estado['puerto']}\n")
        f.write(f"# Inicio      ; {tiempos[0].strftime('%Y-%m-%d %H:%M:%S')}\n")
        f.write(f"# Fin        ; {tiempos[-1].strftime('%Y-%m-%d %H:%M:%S')}\n")
        f.write(f"# Resoluc.   ; {estado['resolucion_s']}\n")
        f.write(f"# Muestras   ; {len(valores)}\n")
        w = csv.writer(f, delimiter=';')
        w.writerow(["Fecha y Hora", "f0 (Hz)"])
        for t, v in zip(tiempos, valores):
            w.writerow([t.strftime('%Y-%m-%d %H:%M:%S'), v])
    print(f"CSV anotado guardado: {ruta}")

btn_csv.on_clicked(exportar_csv)

```

```

# --- Panel de estado ---
ax_estado = fig.add_axes([0.02, 0.04, 0.25, 0.14])
ax_estado.set_facecolor('#f5f5f5')
ax_estado.set_xticks([]); ax_estado.set_yticks([])
for _s in ax_estado.spines.values():
    _s.set_edgecolor('#bbbbbb')
ax_estado.set_title("Estado", fontsize=10, pad=4)
indicador = ax_estado.scatter([0.10], [0.85], s=110, color='cccccc',
                              transform=ax_estado.transAxes,
                              clip_on=False, zorder=3)
texto_estado = ax_estado.text(
    0.22, 0.85, '', fontsize=9.5, va='center', ha='left',
    family='monospace', transform=ax_estado.transAxes,
)
texto_detalle = ax_estado.text(
    0.05, 0.55, '', fontsize=9, va='top', ha='left',
    family='monospace', color='#333333',
    transform=ax_estado.transAxes,
)

def refrescar_panel_estado():
    ahora = datetime.now()
    puerto = estado['puerto'] or '?'
    primer = estado['primer_dato']
    ultima = estado['ultima_lectura']
    msg = estado['mensaje_serial']
    f0 = estado['ultimo_f0']

    if primer is None:
        indicador.set_color('#e0a800')
        encabezado = "Conectando..."
        elapsed = (ahora - estado['tiempo_conexion']).total_seconds()
        detalle = (f"Puerto: {puerto}\n"
                  f"Espera: {elapsed:0.0f} s\n"
                  f"Esperando 1er barrido (~13 s)")
    else:
        indicador.set_color('#28a745')
        encabezado = "Adquiriendo"
        detalle = (f"Puerto: {puerto}\n"
                  f"Última : {ultima.strftime('%H:%M:%S')} if ultima else '--'\n"
                  f"f0 : {f0:.1f} Hz" if f0 else f"Puerto: {puerto}")
        detalle += f"\nResoluc.: {estado['resolucion_s']} s"

    texto_estado.set_text(encabezado)
    if msg:
        detalle += f"\n[Arduino] {msg[:34]}"
    texto_detalle.set_text(detalle)

refrescar_panel_estado()

# --- LÓGICA DE ACTUALIZACIÓN ---
def actualizar(_):
    try:
        linea_serial = arduino.readline().decode('utf-8', errors='ignore').strip()
    except Exception:
        linea_serial = ''

    if linea_serial.startswith('#'):
        msg = linea_serial.lstrip('# ').strip()
        if msg:
            estado['mensaje_serial'] = msg
            refrescar_panel_estado()

```

```

        return linea,

    if not linea_serial.isdigit():
        refrescar_panel_estado()
        return linea,

    f0 = int(linea_serial)
    ahora = datetime.now()
    estado['ultimo_f0'] = float(f0)
    if estado['primer_dato'] is None:
        estado['primer_dato'] = ahora
    estado['ultima_lectura'] = ahora

    paso = estado['resolucion_s']
    if (estado['ultimo_registro'] is None or
        (ahora - estado['ultimo_registro']).total_seconds() >= paso):
        estado['ultimo_registro'] = ahora
        tiempos.append(ahora)
        valores.append(f0)
        with open(ARCHIVO_LOG, 'a', newline='', encoding='utf-8') as f:
            csv.writer(f, delimiter=';').writerow([
                ahora.strftime('%Y-%m-%d %H:%M:%S'), f0, paso
            ])
        print(f"[{paso}s] {ahora.strftime('%H:%M:%S')} f0={f0} Hz")

    if len(tiempos) >= 1:
        linea.set_data(list(tiempos), list(valores))
        if len(tiempos) >= 2:
            ax.set_xlim(tiempos[0], tiempos[-1])
            v_min, v_max = min(valores), max(valores)
            margen = max(5.0, 0.10 * (v_max - v_min))
            ax.set_ylim(v_min - margen, v_max + margen)

    ax.set_title(rf"$f_0$ = {f0} Hz", fontweight='bold')
    fig.autofmt_xdate()
    refrescar_panel_estado()
    return linea,

ani = FuncAnimation(fig, actualizar, interval=200,
                    blit=False, cache_frame_data=False)

try:
    plt.show()
finally:
    try:
        arduino.close()
    except Exception:
        pass

```

Anexo D. Origen de las tablas y las figuras. Todos los datos numéricos y gráficas que aparecen en la sección Resultados y Discusión son directamente derivados de la instrumentación descrita en Metodología (no fueron construidos ni calculados manualmente). El código base de adquisición (Arduino) y monitoreo, registro y

graficación (Python) fue escrito por el profesor Juan Esteban Velez Alvarez (tutor del trabajo de grado), para esta etapa usaremos la captura y visualización de los datos del sensor. Tablas 2 y 3 (datos crudos). Son la salida serial del sensor de resonancia de Helmholtz montado en el Arduino Due (Anexo B). El sketch barrido un rango de frecuencias alrededor del punto de resonancia, encuentra el pico con Goertzel y envía por puerto serie (COM5, baudios 115200) el valor f_0 en Hz una vez por ciclo de barrido. El script de Python (Anexo C) recibe esos valores, los sella con fecha y hora, además, los guardo automáticamente en archivo CSV; ese registro es la fuente exacta de las Tablas 2 y 3. Tabla 1 (estadísticos descriptivos). Se calculó a partir de los valores de las Tablas 2 y 3 (media, desviación estándar poblacional, mínimo, máximo, rango y coeficiente de variación) tal como se explica en el texto de la sección Resultados. Figuras 6 y 7 (evolución temporal f_0). Fueron creadas por la función `construir_figura_publicacion()` del script de Python (Anexo C), que toma la serie temporal registrada en cada prueba y exporta una figura PNG a 600 dpi o PDF vectorial, con los metadatos del ensayo (inicio, fin, duración, media, desviación estándar, mínimo/máximo puerto) incrustados en el propio gráfico. Figura 8 (comparación de las dos pruebas). Se construyó superponiendo los registros CSV exportados de los experimentos con y sin burbujeo generados con el mismo script para compararlos entre sí y frente a las líneas de referencia esperadas para 1 % y 5 % CO₂.