



SOFTWARE DE CÓDIGO ABIERTO PARA EL CONTROL Y MONITOREO DE TRANSDUCTORES ULTRASÓNICOS EN BIOFLUIDOS: ALMACENAMIENTO LOCAL Y PROCESAMIENTO DE DATOS EN TIEMPO REAL

Open-Source Software to Control and Monitor Ultrasonic Transducers in Biofluids: Local Storage and Real-Time Data Processing

Juan Martínez-Escudero 

Universidad Autónoma de Chile (Santiago de Chile, Chile). 

juan.martinez2@cloud.uaautonoma.cl

Romina Muñoz-Buzeta 

Universidad Autónoma de Chile (Santiago de Chile, Chile). 

romina.munoz@uaautonoma.cl

Vicente Salinas-Barrera 

Universidad Autónoma de Chile (Santiago de Chile, Chile). 

vicente.salinas@uaautonoma.cl

Fecha de recibido: 05-08-2024

Fecha de aceptado: 30-12-2024



RESUMEN

El desarrollo de sistemas rápidos y precisos para la caracterización de biofluidos es cada vez más relevante debido a la complejidad de sus cinéticas y las dinámicas moleculares que ocurren en su interior. Estas interacciones reflejan los cambios estructurales de los biopolímeros al interactuar con otros agentes. Monitorear estos procesos es fundamental para comprender los mecanismos subyacentes de dichas interacciones. Este estudio presenta una solución de *software* de código abierto diseñada para controlar y monitorear transductores ultrasónicos para la medición en tiempo real de las propiedades de los biofluidos. El *software*, desarrollado en Python, se integra con *hardware* personalizado para realizar barridos de frecuencia 650 % más rápidos que los dispositivos comerciales como el LCR-8101G, reduciendo el tiempo de adquisición de datos a 0.169 segundos por muestra. Esta mejora significativa permite estudios cinéticos detallados de biofluidos, como se demuestra en la caracterización de las interacciones entre Albúmina Bovina (BSA) y urea. El diseño modular del sistema permite la fácil integración de sensores adicionales, como sondas de temperatura, que pueden mejorar el proceso de caracterización al proporcionar datos más completos. Además, la compatibilidad del *software* con modelos de aprendizaje automático abre la posibilidad de capacidades predictivas avanzadas lo que, potencialmente, permite la identificación en tiempo real de las propiedades de los biofluidos basándose en la respuesta ultrasónica. Los resultados de este estudio, en términos de tiempo de ejecución, validan la eficacia del *software* para acelerar la adquisición de datos sin comprometer la sensibilidad, convirtiéndolo en una herramienta valiosa para los investigadores que estudian fenómenos evolutivos en biofluidos. El trabajo futuro se centrará en expandir aún más las capacidades del sistema, estandarizar el *software* para una adopción más amplia y explorar su aplicación en diversos escenarios experimentales con el propósito de mejorar la comprensión del comportamiento de los biofluidos bajo diferentes condiciones.

Palabras clave: análisis en tiempo real; cinética de biopolímeros; caracterización de biofluidos; respuesta ultrasónica; *software* de código abierto; transductores ultrasónicos.

Cómo citar este artículo: J. Martínez-Escudero, R. Muñoz-Buzeta, y V. Salinas-Barrera, Software de código abierto para el control y monitoreo de transductores ultrasónicos en biofluidos: almacenamiento local y procesamiento de datos en tiempo real. *Revista Facultad de Ingeniería*, vol. 33, no. 70, e18024, 2024. <https://doi.org/10.19053/01211129.v33.n70.2024.18024>

ABSTRACT

The development of fast and precise systems for biofluid characterization has become increasingly important due to the complexity of their kinetics and the dynamic molecular interactions within them. These interactions reveal structural changes in biopolymers when interacting with other agents. Monitoring these processes is essential to understand the underlying mechanisms of such interactions. This study presents an open-source software solution designed to control and monitor ultrasonic transducers and real-time measurement of biofluid properties. It was developed in Python and integrated with custom hardware to perform frequency sweeps 650% faster than commercial devices such as the LCR-8101G; therefore, it reduces data acquisition time to 0.169 seconds per sample. This improvement enables detailed kinetic studies of biofluids, as demonstrated by the characterization of Bovine Serum Albumin (BSA)-urea interactions. The system's modular design allows for easy integration of additional sensors, such as temperature probes, which can enhance the characterization process by providing more comprehensive data. Furthermore, the software's compatibility with machine learning models opens the door to advanced predictive capabilities, potentially allowing for real-time identification of biofluid properties based on ultrasonic response. The findings of this study validate the effectiveness of the software in accelerating data acquisition without compromising sensitivity; hence, it is a valuable tool for researchers studying evolving phenomena in biofluids. Future work will focus on further expanding the system's capabilities, standardizing the software for broader adoption, and exploring its application in various experimental scenarios to enhance the understanding of biofluid behavior under different conditions.

Keywords: biofluid characterization; biopolymer kinetics; open-source software; real-time analysis; ultrasonic response; ultrasonic transducers.

1. INTRODUCCIÓN

El avance en la caracterización de biofluidos ha puesto de manifiesto la importancia de contar con sistemas rápidos y precisos que puedan medir las propiedades físicas de los fluidos de manera no invasiva, ya que, debido a su compleja cinética y a la interacción dinámica entre sus moléculas, se presentan desafíos significativos para lograrla. Esto es especialmente crítico cuando se utilizan transductores ultrasónicos, caso en el que la precisión en la caracterización en frecuencia del transductor es vital para obtener resultados confiables [1]. La capacidad de medir cambios sutiles en las propiedades acústicas del biofluido en tiempo real es esencial para aplicaciones biomédicas, como el monitoreo de procesos terapéuticos y diagnósticos [2-5].

En este contexto, es necesario desarrollar sistemas de caracterización que no solo sean rápidos, sino también adaptables a las restricciones específicas de cada fluido. La frecuencia de resonancia y la respuesta en frecuencia de los transductores ultrasónicos utilizados deben ser evaluadas con alta precisión para garantizar que el sistema se ajuste adecuadamente a la naturaleza del biofluido. Métodos recientes han destacado la relevancia de la sintonización precisa de los transductores para optimizar la transferencia de energía acústica en medios biológicos, lo que requiere herramientas de análisis y control altamente especializadas [6-7]. Además, la adaptabilidad del sistema debe permitir su uso en una variedad de escenarios clínicos y experimentales donde las propiedades del fluido pueden variar considerablemente.

El estudio de la caracterización de transductores ultrasónicos ha sido un área de investigación activa durante varias décadas [8-11]. Estos estudios se centraron en el análisis de la impedancia (Z) que, desde el punto de vista mecánico, da cuenta de la dificultad que tiene el elemento de moverse a una cierta frecuencia. Desde el punto de vista eléctrico, la impedancia será mínima cuando la corriente que fluya por el transductor sea máxima. El otro parámetro de interés en la caracterización es la fase entre la corriente obtenida y el voltaje aplicado, dando cuenta del componente imaginario de la impedancia. Por consecuencia, mecánicamente representa a la energía disipada. Inicialmente, los estudios se centraron

en la caracterización básica de las propiedades eléctricas y mecánicas de los transductores utilizando equipos como analizadores de impedancia o analizadores de espectro que permiten la medición tanto de la impedancia como de la fase en función de la frecuencia.

El trabajo seminal de Berlincourt y Krueger, de 1964, empleó un analizador de impedancia HP 4192A (Hewlett-Packard), con una sensibilidad de 10 m Ω para caracterizar las propiedades eléctricas de transductores piezoeléctricos. Este estudio sentó las bases para futuras investigaciones, permitiendo un entendimiento más profundo de la relación entre la frecuencia de resonancia y la impedancia de los transductores [12].

En los últimos años, la caracterización de transductores ultrasónicos ha evolucionado significativamente con el avance de la tecnología y la disponibilidad de equipos más sensibles y precisos. Riera *et al.* [13] generaron un sistema de caracterización que permite el control y seguimiento de los principales parámetros del transductor: a) tensión y corriente muestreadas en el lado de salida de la unidad de adaptación de impedancias; b) amplitud y fase de vibración muestreadas por un vibrómetro de barrido láser; c) temperatura muestreada por termopar y sonda infrarroja, y d) presión acústica muestreada por micrófono o hidrófono. Con ello, obtuvieron un procedimiento experimental útil para estudiar la estabilidad y la respuesta del transductor durante largos periodos de tiempo, en funcionamiento a alta potencia, así como para determinar el comportamiento histérico (no lineal) de los transductores piezoeléctricos.

Los trabajos recientes se han enfocado en la caracterización de transductores en aplicaciones industriales. Ling *et al.* [14] realizaron el diseño, fabricación y caracterización de un nuevo transductor ultrasónico de alta densidad, utilizando un analizador de impedancia HP 4294A para caracterizar la frecuencia de resonancia y uniformidad del arreglo de transductores. Sus resultados están en buen acuerdo con simulaciones FEM realizadas en el *software* Comsol Multiphysics. En la misma línea industrial, Savoia *et al.* [15] desarrollaron un método preciso para la caracterización de transductores ultrasónicos micromecanizados (CMUTs), utilizando la caracterización de impedancia eléctrica para evaluar parámetros electromecánicos y parasitarios. El método ha sido probado en CMUT acoplados al aire y al agua.

El uso de *software* de código abierto se ha convertido en una solución clave para enfrentar estos desafíos, ya que permite la creación de sistemas personalizables y accesibles para la comunidad científica [16-19]. Los sistemas basados en lenguajes de programación de código abierto, como Python y C++, no solo ofrecen flexibilidad y eficiencia, sino que también facilitan la colaboración entre múltiples usuarios a través de plataformas de control de versiones como Git [20-21]. Este enfoque mejora la reproducibilidad y transparencia de la investigación y acelera el desarrollo y la implementación de nuevas técnicas de caracterización [5-6]. El desarrollo colaborativo y la posibilidad de adaptar el *software* a necesidades específicas de cada experimento o estudio son fundamentales en la evolución de las herramientas para la caracterización de biofluidos.

Sumado a lo anterior, la introducción de lenguajes de programación de código abierto como Python y C++ ha revolucionado la forma en que se realiza la caracterización de transductores ultrasónicos. Python, en particular, ha ganado popularidad debido a su flexibilidad y a la amplia gama de bibliotecas disponibles para el procesamiento de señales y análisis de datos [8-10]. En un estudio reciente, Koerner *et al.* [22] implementaron un sistema de caracterización automatizado utilizando Python, combinando bibliotecas como NumPy y SciPy con *hardware* de adquisición de datos National Instruments PXIe-5162, lo que permitió una caracterización en tiempo real de las propiedades del transductor con una resolución de 16 bits.

En el caso de C++, sigue siendo una opción preferida en aplicaciones donde el rendimiento es crítico. Bayram [23] desarrolló un *software* en C++ para la caracterización de transductores de alta frecuencia, integrado con un analizador de impedancia E4991B. La elección de C++ se motivó por la necesidad de procesamiento de datos en tiempo real con mínima latencia, fundamental para aplicaciones donde el tiempo de respuesta es crítico.

A pesar de los avances en la tecnología de caracterización, existen varias limitaciones. Uno de los principales desafíos es la falta de estandarización en los métodos, lo que dificulta la comparación directa entre estudios. Además, muchos de estos se centran en aplicaciones específicas, limitando la generalización de los resultados a otros contextos.

En este trabajo, se implementó un *software* cuyo objetivo principal es optimizar el tiempo de ejecución, medición, monitoreo y visualización para evitar o minimizar el efecto de la evaporación del fluido durante los experimentos. Este sistema se aplicó con éxito en la caracterización de la cinética BSA interactuando con urea, demostrando su eficacia en estudios donde la preservación de las condiciones del fluido es crucial.

Esta investigación contribuirá a la superación de las limitaciones mencionadas, desarrollando un sistema de caracterización basado en *software* de código abierto, adaptable y estandarizable. El uso de Python permitirá crear un sistema flexible que pueda modificarse según las necesidades de cada experimento. Además, la integración con herramientas de control de versiones como Git facilitará la colaboración y el desarrollo continuo del *software*.

2. MATERIALES Y MÉTODOS

A. Hardware

El sistema de comunicación utilizado en este estudio está compuesto por varios componentes, presentados en la [Figura 1](#).

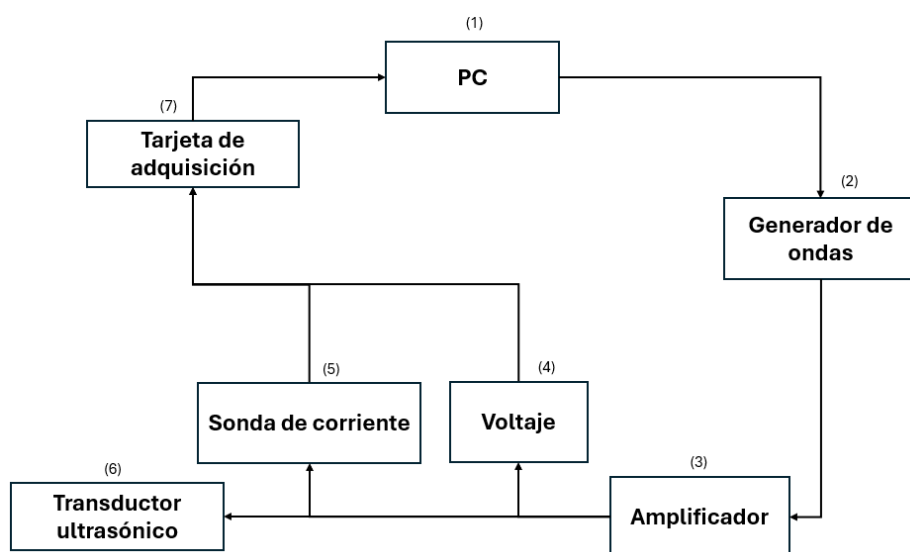


Figura 1. Esquema experimental del sistema de comunicación ultrasónica

El PC (1) controla el generador de ondas, conectado vía USB (2) a través de la interfaz gráfica desarrollada en Python, utilizando PyVisa para ajustar los parámetros de la frecuencia y voltaje. La señal generada se amplifica con el amplificador de potencia (3), antes de ser transmitida por el transductor ultrasónico (6), diseñado bajo los principios de la teoría de Langevin. La corriente que fluye a través del transductor es medida por la sonda de corriente (5) y los datos de voltaje son medidos después de amplificar (4). Esta información es retroalimentada a la PC mediante la tarjeta de adquisición vía PCI-e en sus canales análogos para su posterior procesamiento y análisis.

El PC se encuentra equipado con una tarjeta de adquisición Advantech PCIE-1840L, que cuenta con cuatro canales de entrada analógica con conversión analógica de 16 bit y una velocidad de muestreo de hasta 80 MS/s, permitiendo la captura de señales ultrasónicas. Para la generación de señales, se emplea un generador de funciones Tektronix 33220A capaz de producir señales senoidales, cuadradas y de forma de onda arbitraria, operando a frecuencias de hasta 20 MHz, utilizando técnicas de síntesis digital directa para asegurar la estabilidad de la señal. Esta señal es amplificada mediante un amplificador de potencia SKP MAX300x, que entrega una potencia de salida de hasta 300 W, asegurando su transmisión efectiva a través del transductor ultrasónico. El transductor ultrasónico, desarrollado conforme a la teoría de Langevin, convierte la señal eléctrica amplificada en ondas ultrasónicas mecánicas, necesarias para la realización del experimento. La corriente a través del transductor es monitoreada por una sonda de corriente Pintek PA-699, la cual tiene una sensibilidad de 10 mA/V y es capaz de medir corrientes de hasta 100 A pico.

B. Software

El experimento se desarrolló utilizando Python 3.7 como el lenguaje de programación principal. Para la creación de la interfaz gráfica de usuario (GUI), se implementaron las bibliotecas Tkinter y Custom Tkinter, lo que permitió construir una interfaz interactiva y adaptada, facilitando la interacción del usuario con el *software* y el *hardware* involucrados en el experimento [24-25].

El control y la adquisición de datos del *hardware* se llevaron a cabo mediante DAQ Navi, una biblioteca utilizada para la configuración y lectura de las señales capturadas por la tarjeta de adquisición del sistema. La manipulación y control del generador de ondas se realizó utilizando PyVisa, una biblioteca que permite establecer los parámetros de voltaje y frecuencia necesarios para generar las señales ultrasónicas requeridas durante el experimento.

Para el procesamiento de señales, se utilizaron las bibliotecas NumPy y math, las cuales eran necesarias para realizar cálculos matemáticos y manipular matrices en el análisis de señales. Además, se empleó SciPy específicamente para el filtrado de señales sinusoidales.

El manejo de múltiples hilos de ejecución se gestionó mediante la biblioteca Threading, lo que permitió la realización concurrente de diferentes procesos, asegurando un control coordinado del sistema sin interrupciones. Para medir y gestionar el tiempo de ejecución de las diversas operaciones, se utilizó la biblioteca Time, la cual facilitó la sincronización de las tareas del experimento.

La visualización de los resultados se realizó con la biblioteca Matplotlib que proporcionó las herramientas necesarias para crear gráficos personalizados, facilitando la interpretación y presentación de los datos obtenidos. Además, se empleó Pandas para la organización y exportación de los datos generados durante el experimento en archivos CSV, permitiendo una gestión eficiente de estos.

Para la comunicación con la base de datos local, se implementó Mongoengine, una herramienta que facilita la gestión y manipulación de la base de datos, asegurando que los datos almacenados sean accesibles y estén organizados de manera eficiente.

El entorno de desarrollo se gestionó mediante Anaconda, específicamente Miniconda, una plataforma que facilita la creación y administración de entornos virtuales, garantizando que todas las dependencias y bibliotecas necesarias se instalen y ejecuten en un entorno controlado. Este enfoque asegura la reproducibilidad del experimento y facilita la instalación de paquetes adicionales, conforme lo requiera el desarrollo del proyecto.

Para el almacenamiento de los datos, se optó por utilizar MongoDB, una base de datos no relacional, debido a su capacidad para almacenar documentos de manera eficiente, acorde a los requerimientos del *software* utilizado. Se desarrolló una clase denominada “Experimento” para gestionar el almacenamiento de los datos de cada experimento. Esta clase incluyó los siguientes atributos: nombre (string), fluido (string), sensibilidad (float), pasos (int), frecuencia inicial (float), frecuencia final (float), voltaje (float), comentario (string), impedancia (list), frecuencia (list), desplazamiento de fase (list), repeticiones (int) y tiempo (list).

Además, se implementó una clase adicional denominada “Fluido” para registrar y organizar los datos de los fluidos que se reutilizarían en futuros experimentos, asegurando un manejo estructurado y ordenado de la información.

La [Figura 2](#) presenta un diagrama de flujo del *software* para clarificar lo antes descrito, mostrando la interconexión entre los distintos módulos presentes y utilizados.

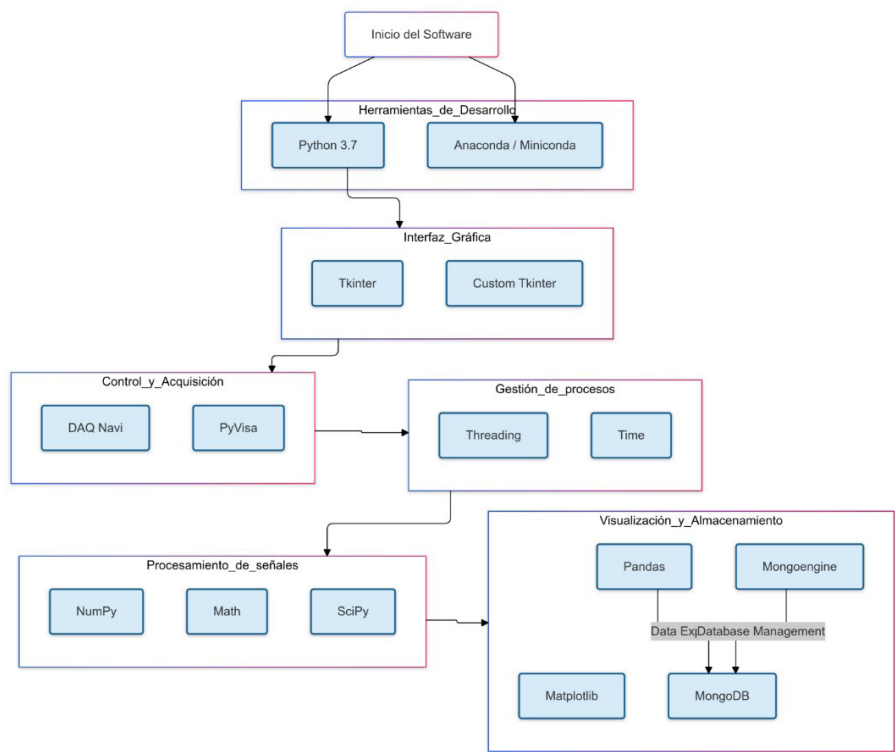


Figura 2. Diagrama de flujo del software generado

C. Análisis de datos

Para calcular la magnitud de la impedancia se utiliza la ley de Ohm como $|z| = V_p/I_p$ donde V_p es el valor peak de la señal de voltaje alterna medida a la frecuencia f , I_p es el valor peak de la señal de corriente alterna medida a la frecuencia f . La señal de voltaje es adquirida por el canal 1 de la tarjeta de adquisición y la corriente por el canal 2.

Para el cálculo de la fase, se realizó una correlación cruzada entre las señales de corriente ($I(t)$) y voltaje ($V(t)$) adquiridas durante el experimento. Esta es una técnica matemática que permite medir la similitud entre dos señales a medida que una se desplaza con respecto a la otra en el tiempo, y permitirá cuantificar el desfase entre ambas cantidades eléctricas.

La correlación cruzada se define matemáticamente como se muestra en la ecuación (1) [26]:

$$R_{IV}(\tau) = \int_{-\infty}^{\infty} I(t) \cdot V(t + \tau) dt \quad (1)$$

donde $R_{IV}(\tau)$ es la función de correlación cruzada que describe la similitud entre las señales de corriente y voltaje desplazadas en el tiempo por un valor τ .

Para implementar esta operación en Python, se utilizó la función “numpy.correlate” de la biblioteca NumPy [27], que hace posible calcular la correlación cruzada entre dos señales discretas. El desfase $\Delta\phi$ entre las señales se determinó identificando el valor de τ que maximiza $R_{IV}(\tau)$, según la siguiente relación que se expresa en la ecuación (2):

$$\Delta\phi = \frac{2\pi \cdot \tau_{m\acute{a}x}}{T} \quad (2)$$

donde T es el período de la señal y $\tau_{m\acute{a}x}$ es el valor de τ correspondiente al máximo de la función de correlación cruzada.

Los datos son adquiridos a frecuencia de muestreo $f_s = 10\text{MHz}$, lo que se traduce en un tiempo de muestreo $t_s = 1/f_s = 1 \cdot 10^{-7}\text{s}$. Con el fin de homologar el cálculo de la correlación para las distintas frecuencias, se impone que se adquieren cinco ciclos acústicos en cada adquisición, independiente de la frecuencia. El largo (l) de la lista que contiene los datos adquiridos para cada frecuencia (f) de excitación, por lo que el tiempo total es $T_{total} = 5/f$. Entonces, el número de datos queda como $l = T_{total}/t_s$.

Se toma el índice del valor más grande del *array* resultante en la correlación cruzada, y se realiza lo expresado en las ecuaciones (3) y (4):

$$T_f = (i - l + 1) \cdot t_s \quad (3)$$

$$Phase\ Value = 2 \cdot \pi \cdot T_f \cdot f_s \quad (4)$$

El resultado de “Phase Value” está en radianes, así que se debe convertir a grados.

Para suavizar las señales de baja amplitud o con una baja relación señal-ruido (SNR), se implementa un filtro Butterworth de orden n , aplicado a la señal de corriente. Este tipo de filtro tiene una respuesta en frecuencia plana en la banda pasante, lo que produce una minimización en la distorsión de la señal resultante. La función de transferencia del filtro está dada por la ecuación (5) [28]:

$$H(s) = \frac{1}{\sqrt{1 + \left(\frac{s}{\omega_c}\right)^{2n}}} \quad (5)$$

donde s es la variable compleja de Laplace, ω_c es la frecuencia de corte, y n es el orden del filtro.

El diseño y la implementación del filtro se realizaron utilizando la biblioteca SciPy, específicamente el submódulo “signal”, que proporciona funciones como “butter” para la creación del filtro y “lfilter” para la aplicación de este a las señales adquiridas [29].

D. Medición en biofluidos

El sistema biofísico utilizado para testear el desempeño del sistema creado corresponde al desplegamiento de la proteína Bovine Serum Albumin (BSA, Merck). Se utiliza una concentración constante de 250 mg/ml de BSA disuelta en agua Milli Q como experimento de control. Para el desplegamiento de la proteína se añade una concentración de 7 M de urea, un agente caotrópico que es capaz de romper los enlaces de la molécula, variando su radio hidrodinámico y, por consiguiente, su viscosidad [30].

Se deposita una gota de 500 de proteína mediante una micropipeta sobre la superficie libre del transductor con el fin de analizar la respuesta en frecuencia de este cuando está en presencia de la proteína en su conformación natural (control) o con urea mientras esta es desplegada. En la [Figura 3](#), se presenta un esquema experimental mostrando el equipamiento utilizado.

Las mediciones fueron realizadas iniciando la excitación del transductor en 29.42 kHz, con una sensibilidad de 0.1 Hz. Se realizaron 1000 pasos hasta alcanzar una frecuencia final de 29.52 kHz.

La respuesta en frecuencia es obtenida inicialmente por un LCR, el cual es capaz de medir inductancia (L), capacitancia (C) y resistencia (R) de un componente electrónico, realizando una variación de la frecuencia de excitación. Luego, fue utilizada la misma configuración experimental con el sistema desarrollado en este trabajo. Ambos experimentos se realizaron en entornos idénticos, donde solo se modificó la metodología y equipamiento para obtener la respuesta en frecuencia del transductor.

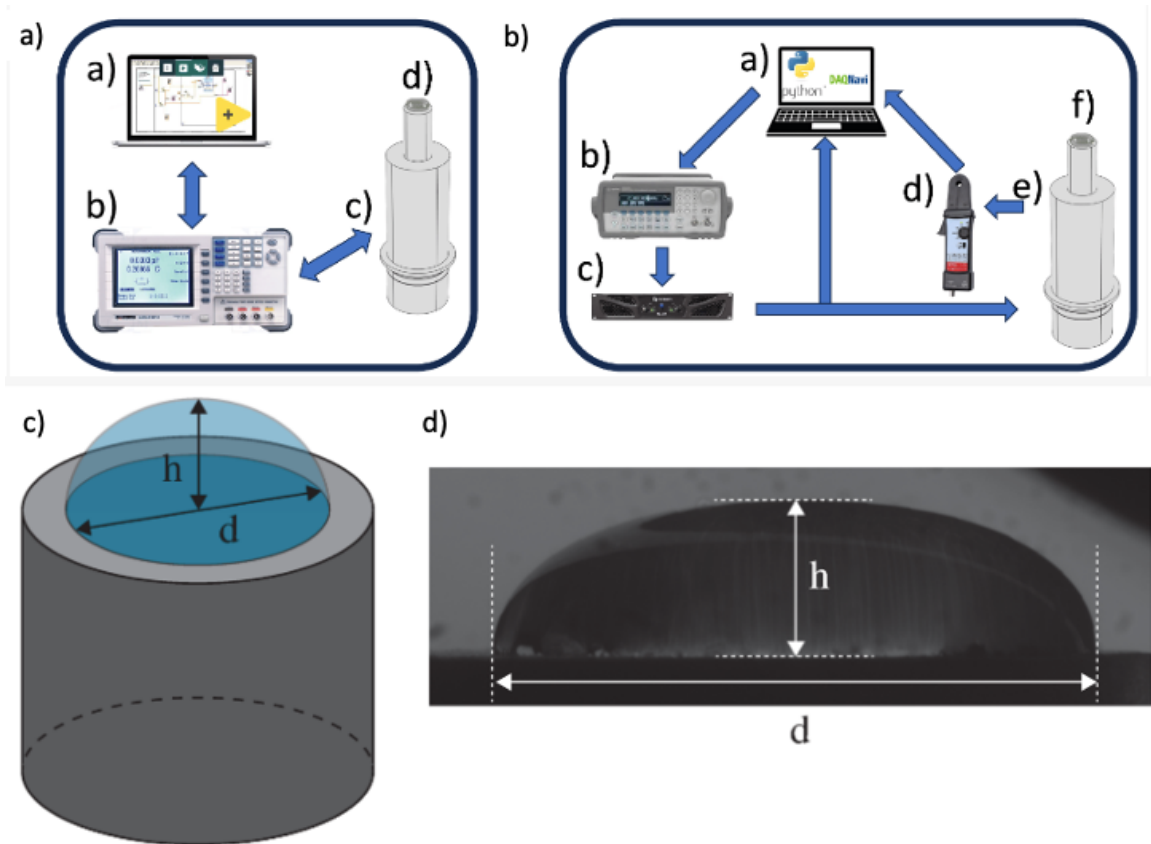


Figura 3. Diagrama esquemático del montaje experimental. A) Mediciones de biofluidos utilizando LCR que contiene (a) computadora con software desarrollado en Labview, (b) medidor LCR GW-Instek, modelo 8101G, (c) transductor ultrasónico resonante de 30 kHz casero basado en el diseño sándwich piezoeléctrico de Langevin y (d) biofluido a caracterizar. B) Mediciones de biofluidos utilizando nuevo sistema experimental que incluye (a) Software Python para controlar el generador de funciones y tarjeta de adquisición de alta velocidad Advantech modelo PCIE-1840L, (b) generador de funciones Agilent modelo 33220A, (c) amplificador de potencia SKP modelo MAX300x para alimentar la señal generada, (d) Sonda de corriente Pintek modelo PA-699 para registrar el flujo de corriente a través del transductor, (e) transductor ultrasónico resonante de 30 kHz de fabricación casera basado en el diseño de sándwich piezoeléctrico de Langevin, (f) biofluido a caracterizar. C) Representación esquemática de la disposición del biofluido sobre el transductor d) imagen de la disposición de la gota de biofluido sobre el transductor.

3. RESULTADOS

La pantalla principal del software desarrollado ofrece acceso a las interfaces gráficas: “Crear experimento pausado”, “Crear experimento”, “Ver datos”, “Historial”, “Configuración” y “Salir”. Para cada interfaz se puede acceder desde la barra lateral deslizante como en la zona central.

Esta interfaz fue diseñada con el objetivo de facilitar la gestión integral de los experimentos, proporcionando accesos directos a las funciones más relevantes. Además, permite la visualización de los datos obtenidos, el acceso a un historial de experimentos anteriores y la configuración del sistema, lo que incluye la selección de dispositivos de *hardware* y la ruta de archivos de configuración.

Este diseño modular y accesible de la interfaz garantiza que los investigadores puedan operar el sistema de manera eficiente, minimizando los errores y maximizando la productividad en el entorno de laboratorio. A su vez, refuerza la importancia de una gestión eficiente del experimento, facilitando el seguimiento de las pruebas realizadas y permitiendo la adaptación del sistema a las necesidades específicas de cada caso.

Esta pantalla permite al usuario seleccionar y configurar el dispositivo de adquisición de datos, el generador de ondas, y establecer la ruta para archivos de configuración en formato XML. Además, ofrece opciones para guardar la configuración actual y cargar una nueva, optimizando la gestión de los parámetros experimentales. Antes de iniciar cualquier experimento, es necesario configurar el sistema y esta información se guarda en la base de datos (Figura 4).

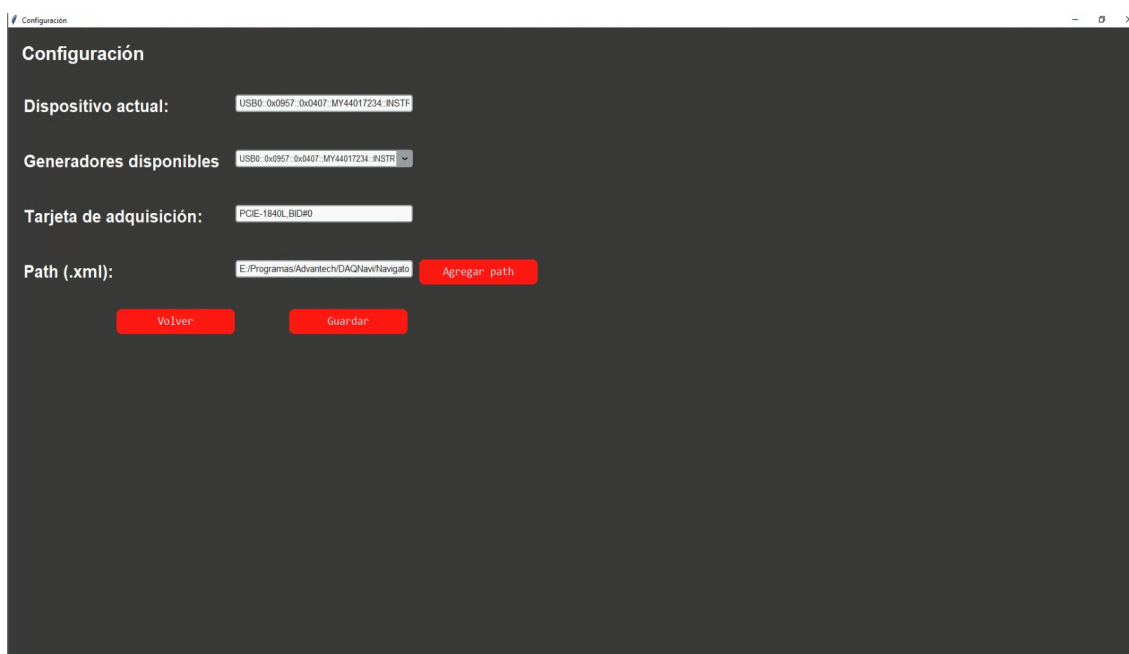


Figura 4. Interfaz de “Configuración” del sistema de caracterización ultrasónica

Los cuatro campos obligatorios que deben configurarse son: “Dispositivo actual”, “Generadores disponibles”, “Tarjeta de adquisición” y “Path (.xml)”. El campo “Generadores disponibles” muestra un listado de los generadores de ondas disponibles conectados vía USB; la flexibilidad del sistema permite controlar múltiples generadores de funciones de la misma marca, seleccionando el equipo deseado.

El campo “Tarjeta de adquisición” es una entrada de texto donde se ingresa el nombre de dispositivo de la tarjeta, mientras que “Path (.xml)” pide seleccionar la ruta con el botón “Agregar path”, donde se ubica este archivo con la configuración de la tarjeta. Este archivo de etiquetado establece las configuraciones específicas de la tarjeta de adquisición para cada medición realizada.

Para visualizar y confirmar si la entrada de los canales es correcta, se tiene la pantalla “Ver datos”, en la que se muestran gráficos en tiempo real utilizando las librerías “Threading” y “Matplotlib” de Python [31].

La [Figura 5](#) muestra la interfaz gráfica creada en Python mediante Tkinter y Custom Tkinter para controlar y monitorear las señales ultrasónicas utilizadas en los experimentos.

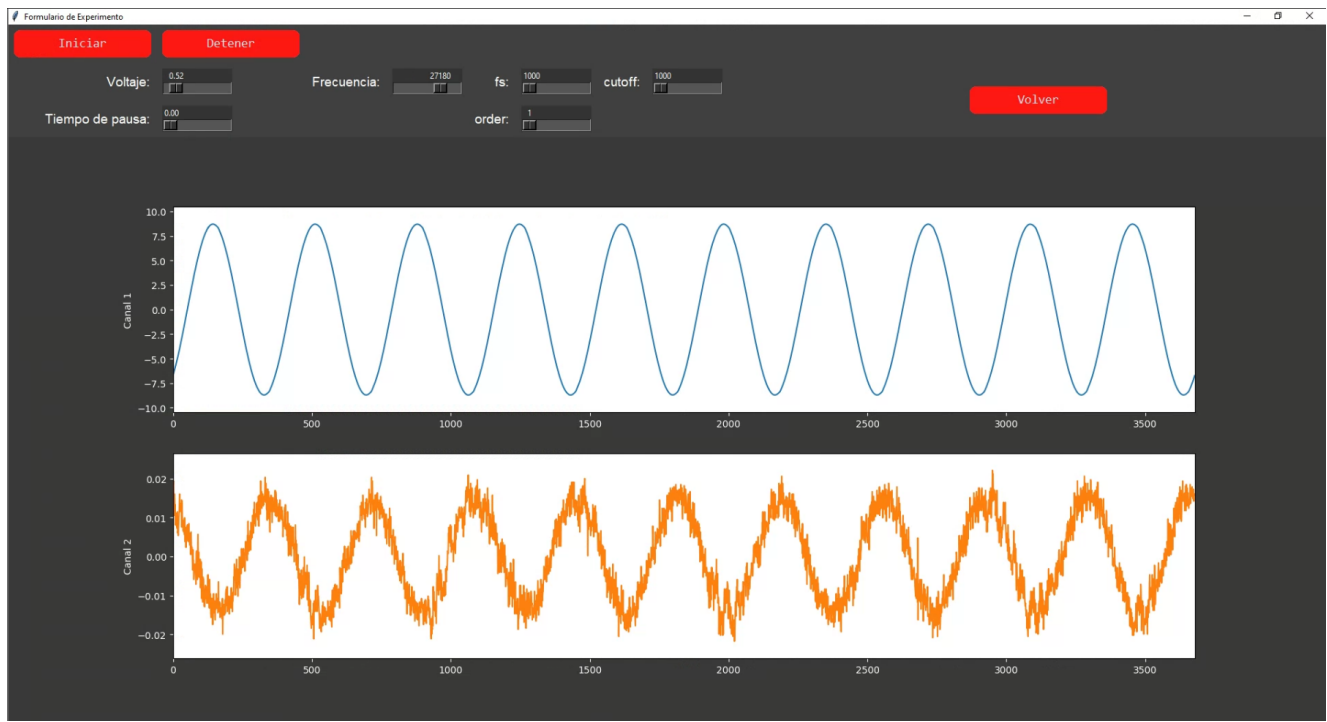


Figura 5. Interfaz gráfica “Ver datos”

Esta interfaz permite la configuración de parámetros como el voltaje de salida del generador de funciones, la frecuencia de la señal ultrasónica y el tiempo de pausa entre las señales. Además, incluye la opción de aplicar el filtro de paso bajo a la señal de corriente capturada por el canal 2 de la tarjeta de adquisición. En la [Figura 6](#), el gráfico superior presenta la señal capturada en el canal 1 sin aplicar ningún filtro, con el fin de asegurar que la señal está siendo adquirida. Por su parte, el gráfico inferior representa la señal del canal 2, después de aplicar el filtro de paso bajo, lo que permite observar el filtrado óptimo, reduciendo el ruido y preservando las características esenciales de la señal.

Este enfoque facilita la evaluación de la efectividad del filtrado en el canal 2 y posibilita definir en tiempo real los parámetros a usar posteriormente en los ensayos experimentales. La función del filtro funciona con tres parámetros descritos en la sección materiales y métodos. En la [Figura 6](#) se muestra el resultado del filtro utilizando los parámetros $n=5$, $\omega_c = 40\text{kHz}$, y una frecuencia de muestreo de $f_s = 10\text{MHz}$.

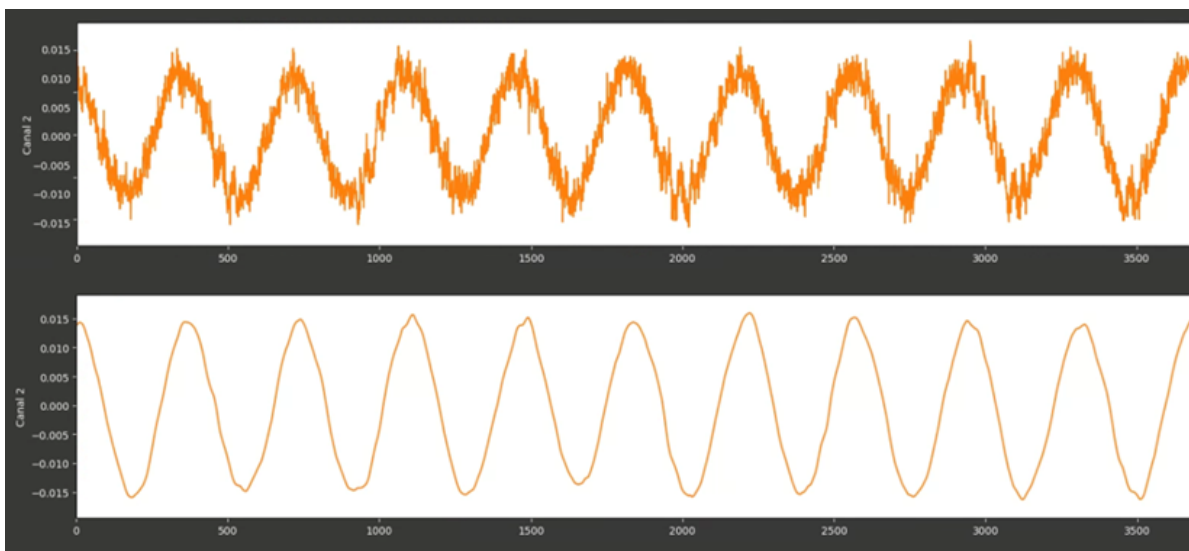


Figura 6. Comparativa entre las señales de corriente aplicando el filtro de Butterworth en el canal 2

A. Experimento continuo

El componente central del *software* es la interfaz gráfica denominada “Experimento continuo”, la cual permite realizar el experimento de manera repetitiva, según el número de repeticiones establecido en el formulario (Figura 7).

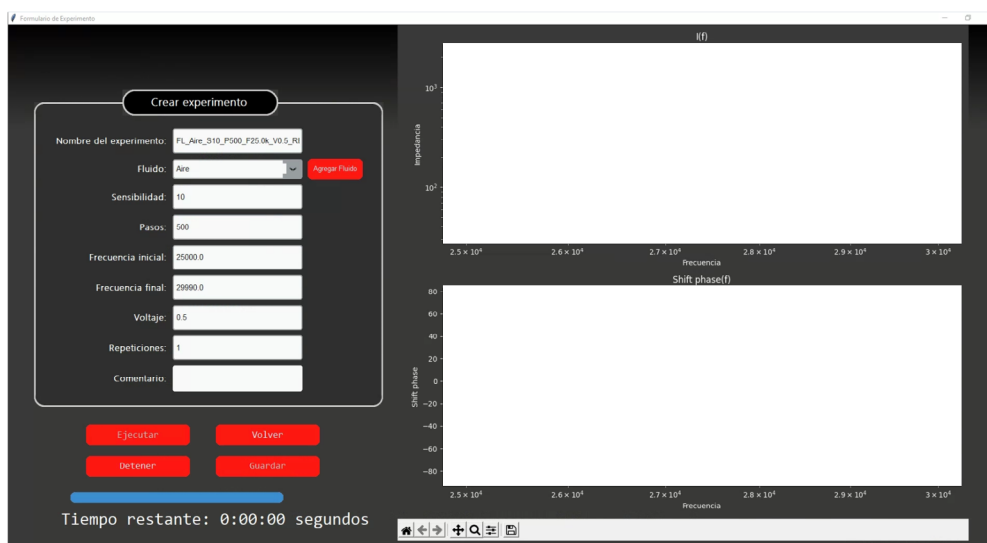


Figura 7. Interfaz para la creación y ejecución de experimentos

La pantalla permite configurar parámetros del experimento como el nombre, fluido utilizado, sensibilidad, número de pasos, frecuencia inicial y final, voltaje y número de repeticiones. Los gráficos a la

derecha están diseñados para mostrar la impedancia y el desfase de la señal en función de la frecuencia, proporcionando una visualización de los resultados experimentales.

Además de las repeticiones, el formulario incluye los campos, previamente mencionados en la clase “Experimento”, para su almacenamiento en la base de datos MongoDB. El usuario puede seleccionar el fluido, creado antes en otra pantalla con los atributos “nombre” y “descripción” y este será almacenado en la base de datos para su uso posterior en el formulario del experimento continuo.

Los campos necesarios en el formulario incluyen el nombre del experimento, que se genera automáticamente a partir de los parámetros configurados, siguiendo una estructura predefinida basada en los datos introducidos en los campos restantes. El campo de “comentario” permite agregar notas adicionales.

A continuación, se especifican los parámetros para la configuración del experimento: la frecuencia inicial (en Hz) desde la cual comenzará la medición, la sensibilidad (incremento por paso) que determinará el aumento en cada iteración y el número de pasos por cada repetición del experimento. El campo de frecuencia final, que no es editable, se genera automáticamente a partir de los valores de los tres campos mencionados. Además, se especifica el voltaje con el cual se ejecutará el experimento y el número de repeticiones que define la cantidad de barridos realizados para el mismo rango de frecuencias.

Una vez seleccionados los campos correspondientes, se presiona el botón “Ejecutar”, lo que activa una alerta para confirmar si se desea proceder con los parámetros configurados. Al presionar “Aceptar”, se inicia el experimento. Durante la ejecución se muestra en pantalla el tiempo restante y una barra de progreso. Al concluir la primera repetición (o al finalizar el experimento si solo se ha seleccionado una repetición), se generan y presentan automáticamente dos gráficos: Z vs f y ϕ vs f .

Al término de la ejecución, se muestra una alerta indicando que el experimento ha finalizado, permitiendo al usuario decidir si desea guardar los resultados en la base de datos. Si se opta por guardar, los datos de impedancia, frecuencia y desplazamiento de fase se almacenan como una lista de listas donde cada sublista representa los resultados de una repetición individual. Tras guardar el experimento, la pantalla se actualiza, los gráficos se borran y el sistema queda preparado para una nueva ejecución.

B. Experimento pausado

En la interfaz “Experimento pausado”, el campo de repeticiones no está presente, ya que no se establece un número fijo de veces para realizar el experimento. El propósito de esta interfaz es obtener un conjunto de datos experimentales correspondientes al mismo sistema con el objetivo de calcular su desviación y estimar el error experimental asociado. Para adquirir cada repetición del experimento, se debe presionar el botón “Siguiente repetición”. Cada respuesta en frecuencia se muestra en pantalla con un color diferente en el gráfico, lo que facilita la identificación de la validez del experimento. La [Figura 8](#) muestra la interfaz correspondiente donde se observa un experimento realizado con dos repeticiones en el mismo sistema experimental, mostradas en azul (primera repetición) y en naranja (segunda repetición).

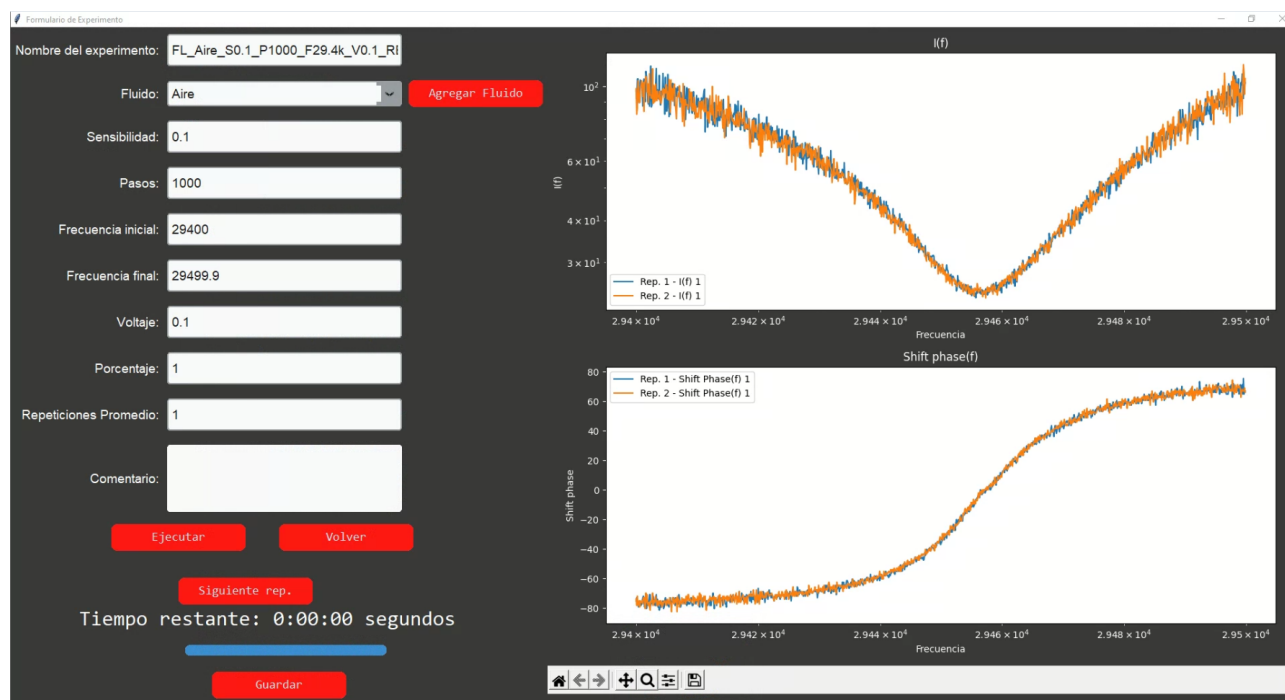


Figura 8. Experimento pausado para un transductor emitiendo en aire con frecuencia inicial de 29.4 kHz, sensibilidad de 0.1 Hz y voltaje de 0.1 V

C. Historial

La Figura 9 presenta la pantalla de historial de experimentos del sistema de caracterización ultrasónica. Esta interfaz está diseñada para proporcionar un acceso eficiente y organizado a los datos de todos los experimentos previamente realizados, centralizando la información clave para su análisis y revisión. La tabla principal de la interfaz incluye columnas que detallan parámetros como el nombre del experimento, el tipo de fluido utilizado, la sensibilidad del sistema, el número de pasos en el barrido de frecuencia, las frecuencias inicial y final, el voltaje aplicado y el número de repeticiones. Esta organización permite a los investigadores comparar diferentes configuraciones experimentales y extraer conclusiones significativas basadas en la variación de estos parámetros.

Además, la interfaz que se implementó con la misma librería de la interfaz gráfica Tkinter y su clase Treeview, cuenta con una serie de botones que permite realizar diversas acciones sobre los experimentos registrados. Entre las funcionalidades disponibles se incluyen la capacidad de ordenar los experimentos según criterios específicos, visualizar gráficos de impedancia (Z) vs. frecuencia (f) y desfase ($\Delta\phi$ vs f), así como exportar estos resultados para análisis posteriores. La capacidad de eliminar experimentos no deseados y exportar datos como $\Delta\phi$ y las frecuencias de calidad, refuerza la utilidad de la interfaz en la gestión de la información experimental.

A diferencia de otras interfaces presentadas en este estudio, esta pantalla se centra en la gestión y análisis de los datos históricos de los experimentos, proporcionando herramientas para la revisión y exportación de información clave relacionada con la caracterización ultrasónica.

Historial

Ver I(f)

Ver Phase(f)

Eliminar

Ordenar por:

Exportar

Volver

Ver Delta Phi

Ver Fr

Ver Z Min.

Ver F. Calidad

Exportar Delta Phi

Exportar Fr

Exportar Z Min.

Exportar F. Calidad

Nombre	Fluido	Sensibilidad	Pasos	Frecuencia Inicial	Frecuencia Final	Voltaje	Repeticiones
Fl_Urea_S0.2_P1000_F25.0k_V0.06_REP1	Urea	0.5	10000	25000.0	20999.5	0.06	1
Fl_Urea_S0.2_P500_F29.22k_V0.06_REP300	Urea	0.2	500	29220.0	29319.8	0.06	300
Fl_Bis_S0.2_P500_F29.2k_V0.06_REP300	Bis	0.2	500	29240.0	29339.8	0.06	300
Fl_Detunación_S0.2_P500_F29.23k_V0.06_REP300	Detunación	0.2	500	29230.0	29329.8	0.06	300
Fl_agua_S.2_P500_F29.27k_V0.06_REP300	agua	0.2	500	29270.0	29369.8	0.06	300
Fl_Detunación_S10_P800_F24.0k_V0.1_REP	Detunación	10.0	800	24000.0	31990.0	0.1	1
Fl_agua_S0.2_P500_F29.2k_V0.06_REP1	agua	0.2	500	29200.0	29299.8	0.1	1
Fl_Bis_S0.2_P500_F29.2k_V0.06_REP200	Bis	0.2	500	29200.0	29299.8	0.06	200
Fl_agua_S0.2_P500_F29.2k_V0.06_REP200	agua	0.2	500	29200.0	29299.8	0.06	200
Fl_Detunación_S0.2_P500_F29.2k_V0.06_REP200	Detunación	0.2	500	29200.0	29299.8	0.06	200
Fl_Bis_S0.2_P500_F29.2k_V0.06_REP20	Bis	0.2	500	29200.0	29299.8	0.06	20
Fl_Bis_S0.2_P500_F29.23k_V0.06_REP20_V2	Bis	0.2	500	29250.0	29349.8	0.06	20
Fl_Bis_S0.2_P500_F29.23k_V0.06_REP20_V1	Bis	0.2	500	29250.0	29349.8	0.06	20
Fl_Bis_S0.2_P500_F29.23k_V0.06_REP20	Bis	0.2	500	29250.0	29349.8	0.06	20
Fl_Urea_S0.2_P500_F29.23k_V0.06_REP20	Urea	0.2	500	29250.0	29349.8	0.06	20
Fl_agua_S0.2_P500_F29.23k_V0.06_REP20_V2	agua	0.2	500	29250.0	29349.8	0.06	20
Fl_agua_S0.2_P500_F29.23k_V0.06_REP20_V1	agua	0.2	500	29250.0	29349.8	0.06	20
Fl_agua_S0.2_P500_F29.23k_V0.06_REP20	agua	0.2	500	29250.0	29349.8	0.06	20
Fl_agua_S0.2_P200_F29.2k_V0.1_REP1	agua	0.2	200	29200.0	29239.8	0.1	3
Fl_agua_S0.2_P200_F29.2k_V0.1_REP2_V3	agua	0.2	200	29200.0	29239.8	0.1	2
Fl_agua_S0.2_P200_F29.2k_V0.1_REP2_V2	agua	0.2	200	29200.0	29239.8	0.1	2
Fl_agua_S0.2_P200_F29.2k_V0.1_REP2_V1	agua	0.2	200	29200.0	29239.8	0.1	2
Fl_agua_S0.2_P200_F29.2k_V0.1_REP1	agua	0.2	200	29200.0	29239.8	0.1	2
Fl_agua_S0.2_P800_F29.2k_V0.1_REP300	agua	0.2	800	29200.0	29359.8	0.1	300
Fl_Aire_S1.0_P1200_F28.4k_V0.5_REP1	Aire	1.0	1200	28400.0	29599.0	0.5	1
Fl_Aire_S1_P1200_F28.4k_V0.5_REP1_V8	agua	1.0	1200	28400.0	29599.0	0.5	2
Fl_Aire_S1_P1200_F28.4k_V0.5_REP1_V5	agua	1.0	1200	28400.0	29599.0	0.5	1
Fl_Aire_S1_P1200_F28.4k_V0.5_REP1_V3	agua	1.0	1200	28400.0	29599.0	0.5	2
Fl_Aire_S1_P1200_F28.4k_V0.5_REP1_V2	agua	1.0	1200	28400.0	29599.0	0.5	1
Fl_agua_S1_P1200_F28.4k_V0.5_REP1	agua	1.0	1200	28400.0	29599.0	0.5	1
Fl_Aire_S1_P1200_F28.4k_V0.5_REP1	agua	1.0	1200	28400.0	29599.0	0.5	1
Fl_agua_S10_P800_F24k_V0.5_REP1	agua	10.0	800	24000.0	31990.0	0.5	1
Fl_Aire_S10_P800_F24k_V0.5_REP1	agua	10.0	800	24000.0	31990.0	0.5	1
Fl_AQUA_S0.5_P000_F1.29.15k_V0.06_2REPS_M	agua	0.5	500	29150.0	29399.5	0.06	None
Fl_AQUA_S0.5_P000_F1.29.15k_V0.06_REP300	agua	0.5	500	29150.0	29399.5	0.06	300
2000 pasos 29.03kHz v0.1 SAMP CORRIENT	agua	0.2	2000	29030.0	29429.8	0.1	1
2000 pasos 29.03kHz v0.06 SAMP CORRIENT	agua	0.2	2000	29030.0	29429.8	0.06	1
2000 pasos 29.03kHz v0.1 SAMP CORRIENT	agua	0.2	2000	29030.0	29429.8	0.1	1
2000 pasos 29.03kHz v0.06 SAMP CORRIENT	agua	0.2	2000	29030.0	29429.8	0.06	1
2000 pasos 29.03kHz v0.1 AMP CORRIENTE	agua	0.2	2000	29030.0	29429.8	0.1	1
2000 pasos 29.03kHz v0.06 AMP CORRIENTE	agua	0.2	2000	29030.0	29429.8	0.06	1

Figura 9. Pantalla de historial de experimentos

D. Exportar datos

En la pantalla “Historial” se ofrece la opción de exportar los conjuntos de datos en formato CSV para un análisis posterior en aplicaciones especializadas, según las necesidades del usuario. La exportación de estos archivos se realiza mediante la biblioteca Pandas de Python, generando archivos con los siguientes encabezados: “Impedancia Rep. n”, “Fase Rep. n” y “Frecuencia Rep. N”, donde *Rep. n* corresponde al número de la repetición. Las columnas generadas tendrán la misma longitud y el número de filas estará limitado por la cantidad de pasos realizados en el experimento.

Es importante señalar que, para poder exportar los datos, es necesario haber visualizado previamente el gráfico del experimento en cuestión (los gráficos de impedancia y fase se registran automáticamente durante este proceso). Esto se debe a que, al mostrar el gráfico, se guarda un registro en la base de datos con los cálculos ya realizados, lo que permite que, al volver a visualizarlo, no sea necesario repetir todos los cálculos, optimizando el rendimiento computacional de las funcionalidades “Ver experimento” y “Exportar experimento”.

Para almacenar estos registros, se crean clases en MongoDB correspondientes a cada categoría de postprocesamiento. Cada documento en estas clases incluye tres atributos: “nombre” (tipo string), “datos” (tipo array) y “tiempo” (tipo array), lo que facilita la organización y el acceso eficiente a los datos procesados.

4. ANÁLISIS

Para analizar el comportamiento del sistema generado se realizó la caracterización de un transductor ultrasónico con una frecuencia de resonancia de 30 kHz. En este análisis, se buscó identificar la zona donde se minimiza la impedancia, característica fundamental para determinar la frecuencia de resonancia del transductor. Además, en este punto de resonancia, se observó una marcada variación en la fase, confirmando la correcta identificación de la frecuencia resonante.

Los resultados obtenidos sin la aplicación de fluido en la cara radiante del transductor muestran que tanto la impedancia como la fase se comportan de acuerdo con las expectativas, sin resultados inesperados. La impedancia se mantuvo en torno al valor esperado y las cantidades Z (impedancia) y $\Delta\Phi$ (fase) fueron consistentes con lo anticipado.

En la [Figura 10](#) se muestra la pantalla del “Experimento continuo” donde se observa la medición del transductor sin la presencia de fluido, utilizando una frecuencia inicial de 25 kHz, una sensibilidad de 10 Hz y 1000 pasos para alcanzar una frecuencia final de 30 kHz. La resonancia del transductor se identifica claramente en el rango entre 29 kHz y 30 kHz donde la impedancia alcanza su mínimo.

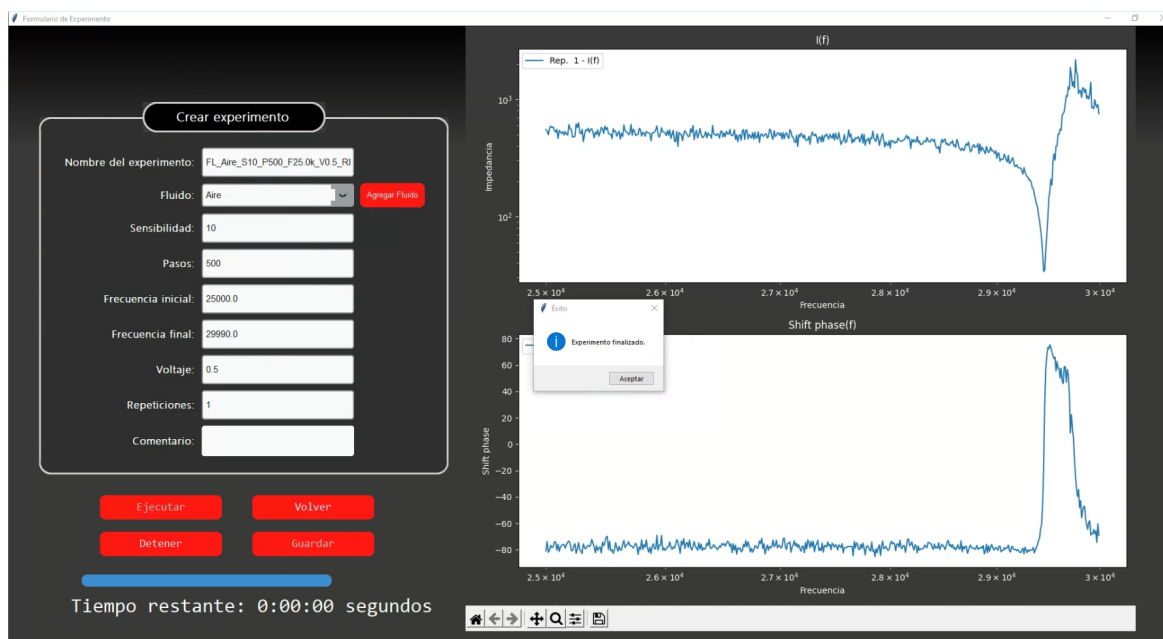


Figura 10. Ejemplo de experimento en aire con una frecuencia inicial de 25 kHz, sensibilidad de 10 Hz y voltaje de 0.5 V

Por su parte, en la [Figura 11](#) se presenta un análisis detallado de la respuesta en frecuencia del sistema, enfocándose en un rango estrecho alrededor de la frecuencia de resonancia. Para este experimento, se configuró una frecuencia inicial de 29.42 kHz, con una sensibilidad de 0.1 Hz y se realizaron 1000 pasos hasta alcanzar una frecuencia final de 29.52 kHz. Este intervalo fue seleccionado específicamente para capturar con alta precisión la zona de resonancia del transductor ultrasónico.

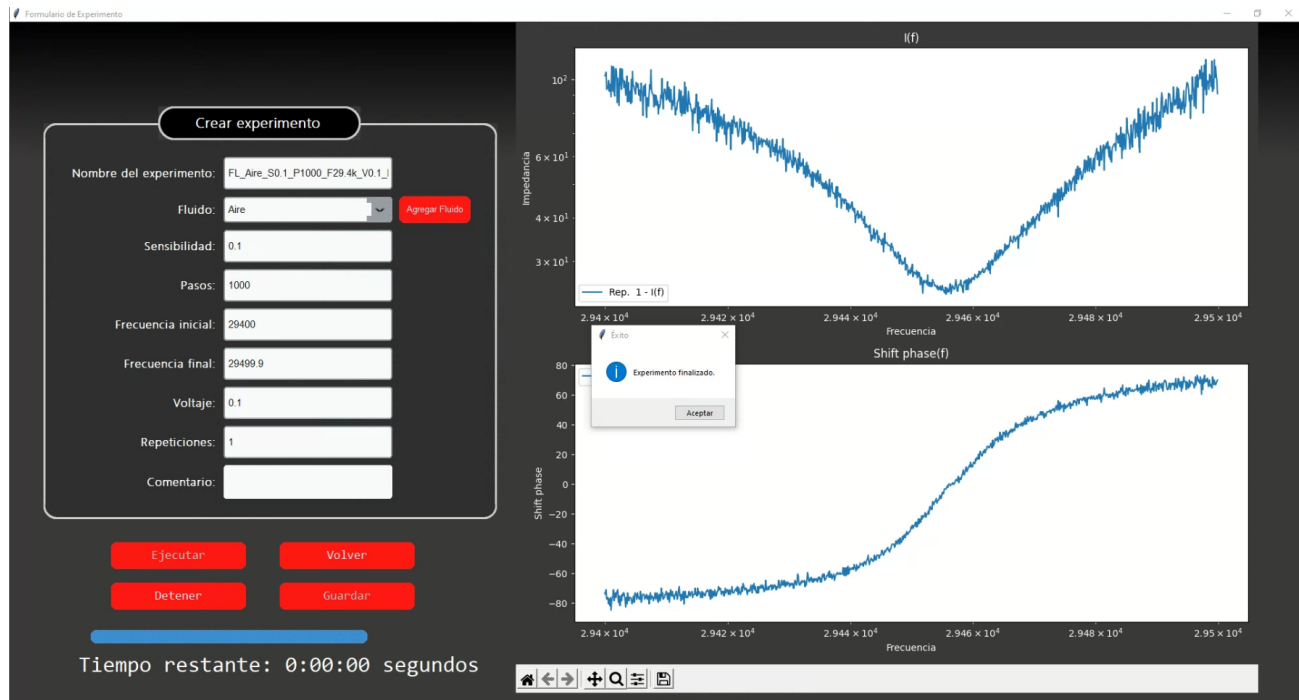


Figura 11. Ejemplo de experimento en aire con una frecuencia inicial de 29.4 kHz, una sensibilidad de 0.1 Hz y un voltaje de 0.1 V

Los datos del canal 2 fueron filtrados utilizando un filtro Butterworth con los parámetros previamente establecidos. El gráfico muestra un enfoque detallado (zoom) en torno a la frecuencia de resonancia obtenida tras 1000 pasos hasta alcanzar una frecuencia final de 29.5 kHz.

El gráfico ilustra cómo, a medida que se avanza en los pasos, la impedancia disminuye hasta alcanzar un mínimo en la frecuencia de resonancia, lo que es característico del comportamiento resonante del transductor. A su vez, se observa la respuesta de fase, que también muestra una transformación significativa en esta región cambiando de signo y, por ende, pasando por desfase igual a cero, indicando la transición a través del punto de resonancia.

Los datos presentados en la [Figura 11](#) provienen del cálculo de impedancia. Para esto, los datos del canal 2 se filtraron utilizando un filtro Butterworth, lo que mejoró la claridad de la señal al eliminar el ruido de alta frecuencia que podría interferir con la precisión de la medición. Esta filtración es especialmente relevante en experimentos como este, donde la sensibilidad del sistema es alta y se requiere una señal limpia para obtener mediciones precisas de la frecuencia de resonancia.

Así pues, se confirmó la eficacia del filtro Butterworth para mejorar la calidad de la señal en el canal 2, facilitando la identificación precisa de la frecuencia de resonancia. Este hallazgo es consistente con los resultados obtenidos en experimentos previos realizados en condiciones similares [1].

En conjunto, los datos presentados validan la configuración del sistema y el enfoque metodológico empleado, demostrando su capacidad para realizar mediciones detalladas y precisas en el entorno experimental propuesto.

A. Tiempos de respuesta

Uno de los desafíos fundamentales en la caracterización de materiales, especialmente en el estudio de biofluidos, es la capacidad de registrar rápidamente los cambios que ocurren cuando estos sistemas interactúan con otros agentes. Este desafío se intensifica cuando se busca mantener una alta sensibilidad en los resultados, comparable con las soluciones disponibles en el mercado, mientras se acelera el tiempo de adquisición para capturar la cinética de los fluidos que evolucionan rápidamente con el tiempo.

Para medir los tiempos de respuesta del sistema, se utilizó la biblioteca Time de Python, definiendo la variable “inicio” antes de ejecutar la función y la variable “final” después de la ejecución de la función. De esta forma, el tiempo de ejecución fue determinado como $\Delta t = \text{final} - \text{inicio}$

Los resultados obtenidos muestran un tiempo promedio de 15.39 segundos por cada set de datos, donde cada set considera 500 puntos en la curva de respuesta en frecuencia del transductor (impedancia versus frecuencia y fase versus frecuencia), esto se traduce en aproximadamente 0.03078 segundos por dato. Cabe destacar que para la obtención de los valores de impedancia y fase se adquiere la señal de voltaje y corriente de cinco ciclos acústicos a una frecuencia de adquisición de 10 MHz.

Las curvas de respuesta en frecuencia de un transductor pueden ser adquiridas mediante un medidor LCR, realizando una variación de la frecuencia de excitación. Utilizando esta metodología, las mediciones de cada set de datos para el mismo transductor, y considerando el mismo número de puntos, son realizadas en aproximadamente 100 segundos [1], lo que indica una mejora de 6.5 veces en la rapidez de adquisición.

Es importante destacar que, durante el cambio de frecuencia en cada muestra, según la sensibilidad configurada, se detuvo la adquisición de datos para modificar la frecuencia en el generador de ondas antes de reanudar la adquisición. Aunque esta combinación de bibliotecas resultó efectiva, se identificó una limitación: la necesidad de pausar la adquisición de datos para evitar su acumulación en el buffer, lo que podría afectar la precisión de los valores obtenidos.

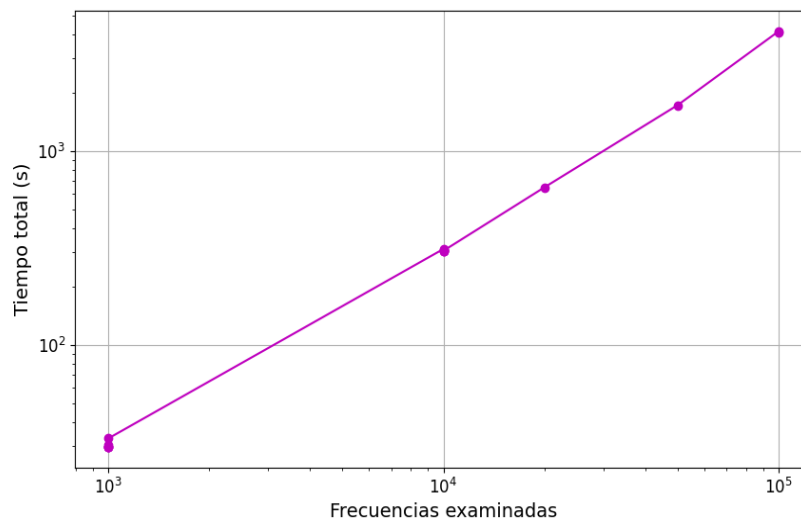
La mayor parte del tiempo de ejecución se destinó a iniciar y detener la adquisición de datos, con un promedio de 4.09 segundos para iniciar y 9.04 segundos para detener, en un experimento de 500 muestras. El tiempo dedicado exclusivamente a la toma de datos, es decir, después de iniciar y antes de detener la adquisición, fue de 1.83 segundos en promedio. Dado que el tiempo total de ejecución por repetición es de 15.39 segundos, se estima que 0.86 segundos se asocian a otros procesos. Esto significa que aproximadamente el 85.4 % del tiempo de ejecución se dedica a las operaciones de inicio y detención de la adquisición.

En la [Tabla 1](#) se presentan los resultados referentes al tiempo de ejecución del *software* para diversas frecuencias de excitación dentro del mismo espectro de barrido. Se observa que el tiempo de ejecución aumenta linealmente con el número de frecuencias exploradas. Sin embargo, este es independiente de la sensibilidad o resolución en el paso de frecuencia.

Tabla 1. Número de frecuencias exploradas en el mismo rango, variando sensibilidad y midiendo el tiempo de respuesta del sistema

# Frecuencias excitación (10^3)	Sensibilidad (kHz)	Frecuencia Inicial (kHz)	Frecuencia Final (kHz)	Tiempo ejecución (s)
1	5.00	25	30	30.36
10	0.50	25	30	306.84
20	0.25	25	30	647.54
50	0.10	25	30	1720.20
100	0.05	25	30	4109.83

La [Figura 12](#) muestra un crecimiento logarítmico del tiempo total de ejecución a medida que aumenta el número de frecuencias examinadas, evidenciando la relación directa entre la cantidad de datos procesados y el tiempo requerido para completar el experimento. Para aumentar el número de frecuencias exploradas se modificó el tamaño del paso de avance en la medición (aumento la sensibilidad) entre 0.05 Hz y 5 Hz.

**Figura 12.** Relación entre el número de frecuencias examinadas y el tiempo total de ejecución en un experimento de caracterización ultrasónica

A su vez, la [Figura 12](#) ilustra la relación entre el número de frecuencias examinadas y el tiempo total de ejecución en un experimento de caracterización ultrasónica. En el eje horizontal se representa el número de frecuencias examinadas en una escala logarítmica, mientras que en el eje vertical se muestra el tiempo total de ejecución en segundos, también en escala logarítmica. La gráfica indica un comportamiento lineal en esta escala logarítmica. Esto sugiere que el tiempo total requerido para completar el experimento aumenta proporcionalmente con el número de frecuencias que se analizan.

Este comportamiento es consistente con la naturaleza del experimento, donde cada frecuencia examinada requiere un procesamiento y análisis específicos, lo que se traduce en un incremento acumulativo del tiempo total a medida que se amplía el rango de frecuencias evaluadas. La pendiente de la línea proporciona información sobre la eficiencia del sistema y la escalabilidad del proceso experimental cuando se incrementa el número de frecuencias.

En comparación con otras figuras presentadas en el estudio, esta gráfica permite comprender las limitaciones temporales del sistema al realizar barridos de frecuencia amplios.

Estos resultados destacan la eficiencia del *software* desarrollado en términos de velocidad y manejo de recursos, cumpliendo con el objetivo de acelerar la adquisición de datos sin comprometer la sensibilidad necesaria para la caracterización precisa de los biofluidos.

B. Experimento en continuo

Para ejemplificar la capacidad del *software* desarrollado, se hace referencia al artículo ya publicado, “Ultrasonic Sensor: A Fast and Non-Destructive System to Measure the Viscosity and Density of Molecular Fluids” [1], en el cual se aplicó la metodología descrita en este estudio para caracterizar la interacción cinética de la proteína BSA con urea. En dicho trabajo, se implementó todo el procedimiento tanto de *hardware* como de *software* detallado en este artículo. La aplicación de este sistema permitió comprender el comportamiento de la viscosidad y la densidad de distintos fluidos bajo condiciones específicas de presión y temperatura.

En este contexto, se realizaron 300 repeticiones para analizar la respuesta en frecuencia del transductor a lo largo del tiempo. El *software* desarrollado demostró ser una herramienta eficaz para explorar sistemas dinámicos, permitiendo capturar la evolución temporal del sistema con una alta tasa de refresco.

La [Figura 13](#) presenta los resultados obtenidos al analizar la evolución temporal de sistema gota-transductor para diferentes fluidos y biofluidos; al seleccionar y visualizar los datos almacenados en la base de datos, y al presionar el botón “Ver Z(f)”, se despliega un gráfico generado con la librería Matplotlib, donde se muestra la respuesta en frecuencia del transductor para las 300 repeticiones. El mapeo de color, que varía del azul (para las primeras repeticiones) al rojo (para las últimas), permite identificar visualmente la evolución temporal del sistema.

En el caso de la Figura 14 d), la cinética de desplegamiento esperada para el BSA al añadir urea, demuestra una disminución de la impedancia mínima en el tiempo, además del movimiento de la frecuencia de resonancia, la cual es atribuida a la evaporación de la gota sobre la superficie del transductor y la sensibilidad del sistema a las variaciones de la respuesta en frecuencia [1]. Por su parte, la variación de impedancia puede correlacionarse con los cambios en la viscosidad del biofluido, los cuales son consecuencia de la interacción molecular entre las moléculas de BSA y urea. A medida que la urea despliega las moléculas de BSA, se observa un aumento en el volumen efectivo ocupado por estas en el fluido, lo que se traduce en un aumento de viscosidad del sistema, produciendo variaciones de la respuesta en frecuencia [1, 32, 33].

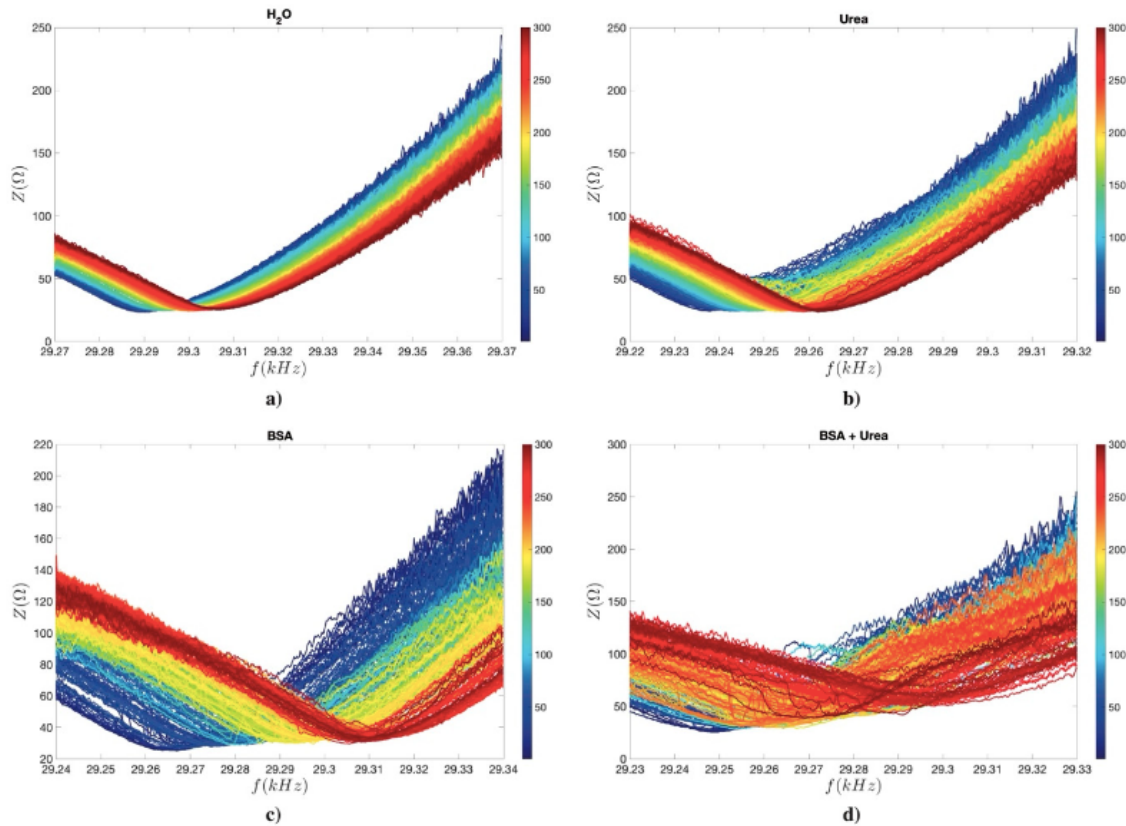


Figura 13. Señales adquiridas durante el proceso de caracterización de diferentes fluidos y biofluidos. Nota: a) control con agua MilliQ; b) control con urea a 7M; c) control con BSA; d) desplegamiento de BSA+urea.

5. CONCLUSIONES Y TRABAJOS FUTUROS

El *software* desarrollado demuestra ser una herramienta eficaz y robusta para la caracterización rápida y precisa de biofluidos utilizando transductores ultrasónicos. La capacidad del sistema para realizar barridos de frecuencia a una velocidad 650 % superior en comparación con dispositivos comerciales como el LCR-8101G, con una mejora equivalente a 0.169 segundos por muestra, resalta su potencial para aplicaciones donde la rapidez en la adquisición de datos es crucial.

Este avance no solo facilita la caracterización en tiempo real de biofluidos, sino que también abre nuevas posibilidades para su aplicación en investigaciones biomédicas y de materiales donde la evolución temporal de las propiedades del fluido es de interés crítico. Cabe destacar que los tiempos de respuesta antes obtenidos se han logrado con sensibilidad de 0.1 Hz de frecuencia, 50 mΩ de sensibilidad en impedancia y 0.2 (deg) de sensibilidad de fase, transformándose en un dispositivo de caracterización de alta sensibilidad experimental.

Sumado a lo anterior, la implementación del sistema descrito, incluyendo *hardware* y *software*, permite optimizar los tiempos de adquisición y expandir las capacidades del sistema, posibilitando la incorporación de sensores adicionales, como los de temperatura, que pueden proporcionar una caracterización más completa del sistema. La recopilación y análisis de estos datos en conjunto ofrece la oportunidad de desarrollar modelos avanzados de *machine learning* que podrían predecir las propiedades

del fluido basándose en la respuesta del sistema ultrasónico, lo que contribuiría significativamente a la automatización y mejora de la precisión en la caracterización de biofluidos.

En trabajos futuros, se podrá explorar la implementación de estas mejoras, así como la validación del sistema en diferentes escenarios experimentales con el objetivo de consolidar su uso en la investigación avanzada de biofluidos y materiales. La estandarización del *software* y su código abierto también favorecerá una adopción más amplia en la comunidad científica, promoviendo la colaboración y el desarrollo continuo de nuevas técnicas y aplicaciones.

CONTRIBUCIÓN DE AUTORÍA

Juan Martínez-Escudero: Desarrollo de software; Adquisición de datos; Visualización.

Romina Muñoz-Buzeta: Investigación; Procesamiento de datos; Análisis formal; Redacción-Borrador original.

Vicente Salinas-Barrera: Conceptualización; Investigación; Análisis formal; Redacción - revisión y edición.

REFERENCIAS

- [1] R. Muñoz *et al.*, "Ultrasonic Sensor: A Fast and Non-Destructive System to Measure the Viscosity and Density of Molecular Fluids," *Biosensors (Basel)*, vol. 14, no. 7, e346, Jul. 2024.
<https://doi.org/10.3390/bios14070346>
- [2] M. Evander *et al.*, "Noninvasive Acoustic Cell Trapping in a Microfluidic Perfusion System for Online Bioassays," *Analytical Chemistry*, vol. 79, no. 7, pp. 2984-2991, Apr. 2007.
<https://doi.org/10.1021/ac061576v>
- [3] B. Godber *et al.*, "Direct Quantification of Analyte Concentration by Resonant Acoustic Profiling," *Clinical Chemistry*, vol. 51, no. 10, pp. 1962-1972, Oct. 2005.
<https://doi.org/10.1373/clinchem.2005.053249>
- [4] F. Sarry, M. I. Rocha-Gaso, A. Renaudin, D. Beyssen, M. Oudich, P. Charette, "In situ characterization of biofluid using an optimized hybrid acousto-optic sensor array on a microfluidic cell," in *IEEE Sensors*, Oct. 2017, pp. 1-3. <https://doi.org/10.1109/ICSENS.2017.8234344>
- [5] P. Li, T. J. Huang, "Applications of Acoustofluidics in Bioanalytical Chemistry," *Analytical Chemistry*, vol. 91, no. 1, pp. 757-767, Jan. 2019. <https://doi.org/10.1021/acs.analchem.8b03786>
- [6] Y. Hou, F. Mao, Y. Wu, Y. Zhu, H. Zhang, "Broadband tunable acoustic impedance matching using gradient-distributed piezoelectric structure," *Applied Physics Letters*, vol. 124, no. 15, e203019, Apr. 2024. <https://doi.org/10.1063/5.0203019>
- [7] V. Ostasevicius *et al.*, "Development of a Low-Frequency Piezoelectric Ultrasonic Transducer for Biological Tissue Sonication," *Sensors*, vol. 23, no. 7, e3608, Mar. 2023.
<https://doi.org/10.3390/s23073608>
- [8] C. Liu, F. T. Djuth, Q. Zhou, K. K. Shung, "Micromachining techniques in developing high-frequency piezoelectric composite ultrasonic array transducers," *IEEE Transactions on Ultrasonics Ferroelectrics*

- and Frequency Control, vol. 60, no. 12, pp. 2615-2625, Dec. 2013.
<https://doi.org/010.1109/TUFFC.2013.2860>
- [9] P. Muralt et al., "Piezoelectric micromachined ultrasonic transducers based on PZT thin films," *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 52, no. 12, pp. 2276-2288, Dec. 2005. <https://doi.org/10.1109/TUFFC.2005.1563270>
- [10] F. Akasheh, J. D. Fraser, S. Bose, A. Bandyopadhyay, "Piezoelectric micromachined ultrasonic transducers: modeling the influence of structural parameters on device performance," *IEEE Transactions on Ultrasonics Ferroelectrics and Frequency Control*, vol. 52, no. 3, pp. 455-468, Mar. 2005. <https://doi.org/10.1109/TUFFC.2005.1417268>
- [11] D. D. Bennink, "Characterization of Ultrasonic Immersion Transducers," in *Review of Progress in Quantitative Nondestructive Evaluation*, Boston, MA: Springer US, 1990, pp. 781-788.
https://doi.org/10.1007/978-1-4684-5772-8_98
- [12] S. Takahashi, Y. Sasaki, M. Umeda, K. Nakamura, S. Ueha, "Nonlinear behavior in piezoelectric ceramic transducers," in *12th IEEE International Symposium on Applications of Ferroelectrics*, 2000, pp. 11-16. <https://doi.org/10.1109/ISAF.2000.941503>
- [13] E. Riera, J. A. Gallego-Juárez, A. Blanco, V. M. Acosta, "Power characterization of ultrasonic piezoelectric transducers," in *Proceedings of the International Congress on Ultrasonics*, Vienna University of Technology, 2007. https://doi.org/10.3728/ICUltrasonics.2007.Vienna.1435_riera
- [14] J. Ling et al., "Design and Characterization of High-Density Ultrasonic Transducer Array," *IEEE Sensors Journal*, vol. 18, no. 6, pp. 2285-2290, Mar. 2018. <https://doi.org/10.1109/JSEN.2018.2797222>
- [15] A. S. Savoia, B. Mauti, G. Caliano, "Accurate evaluation of the electro-mechanical and parasitic parameters of CMUTs through electrical impedance characterization," in *IEEE International Ultrasonics Symposium (IUS)*, IEEE, Sep. 2017. <https://doi.org/10.1109/ULTSYM.2017.8092910>
- [16] J. M. Pearce, "The Future of Open-Source Hardware and Science," in *Open-Source Lab*, Elsevier, 2014, pp. 255-263. <https://doi.org/10.1016/B978-0-12-410462-4.00007-X>
- [17] J. M. Pearce, "Building Research Equipment with Free, Open-Source Hardware," *Science* (1979), vol. 337, no. 6100, pp. 1303-1304, Sep. 2012. <https://doi.org/10.1126/science.1228183>
- [18] C. Zhang, N. C. Anzalone, R. P. Faria, J. M. Pearce, "Open-Source 3D-Printable Optics Equipment," *PLoS One*, vol. 8, no. 3, e59840, Mar. 2013. <https://doi.org/10.1371/journal.pone.0059840>
- [19] R. L. Tanner, K. L. Onthank, "Open Source Solutions in Experimental Design: An Introduction to the Symposium," *Integrative and Comparative Biology*, vol. 62, no. 4, pp. 1056-1060, Oct. 2022.
<https://doi.org/10.1093/icb/icac132>
- [20] G. J. Stallman, *Free Software, Free Society: Selected Essays of Richard M. Stallman*. CreateSpace, 2009.
- [21] F. L. van Rossum, *The Python Language Reference Manual*. Network Theory Ltd., 2011.
- [22] L. J. Koerner, T. A. Caswell, D. B. Allan, S. I. Campbell, "A Python Instrument Control and Data Acquisition Suite for Reproducible Research," *IEEE Transaction on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1698-1707, Apr. 2020. <https://doi.org/10.1109/TIM.2019.2914711>
- [23] B. Bayram, "Diamond-based capacitive micromachined ultrasonic transducers," *Diamond Related Materials*, vol. 22, pp. 6-11, Feb. 2012. <https://doi.org/10.1016/j.diamond.2011.11.006>

- [24] B. M. Pardi *et al.*, “pyQCM-BraTaDio: A tool for visualization, data mining, and modelling of Quartz crystal microbalance with dissipation data,” *The Journal of Open Source Software*, 2023.
<https://doi.org/10.1101/2023.12.15.571789>.
- [25] A. Alsabbagh, A. N. Nasser, G. Husi, “Python-Based Data Analysis Tool for a 6-DoF Industrial Robot,” *IOP Conference Series: Material Science and Engineering*, vol. 568, no. 1, e012098, Aug. 2019.
<https://doi.org/10.1088/1757-899X/568/1/012098>
- [26] R. Shiavi, “Theory and application of cross correlation and coherence,” in *Introduction to Applied Statistical Signal Analysis*, Elsevier, 2007, pp. 331-366.
<https://doi.org/10.1016/B978-012088581-7/50026-1>
- [27] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357-362, Sep. 2020. <https://doi.org/10.1038/s41586-020-2649-2>
- [28] N. Gilbert, J. Fleming, “Direct form expansion of the transfer function for a digital Butterworth low-pass filter,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 30, no. 6, pp. 1004-1006, Dec. 1982. <https://doi.org/10.1109/TASSP.1982.1163993>
- [29] P. Virtanen *et al.*, “SciPy 1.0: fundamental algorithms for scientific computing in Python,” *Nature Methods*, vol. 17, no. 3, pp. 261-272, Mar. 2020. <https://doi.org/10.1038/s41592-019-0686-2>
- [30] J. A. Jones, D. K. Wilkins, L. J. Smith, C. M. Dobson, “Characterisation of protein unfolding by NMR diffusion measurements,” *Journal of Biomolecular NMR*, vol. 10, no. 2, pp. 199-203, 1997.
<https://doi.org/10.1023/A:1018304117895>
- [31] J. A. Izaac, J. B. Wang, “pyCTQW: A continuous-time quantum walk simulator on distributed memory computers,” *Computer Physics Communication*, vol. 186, pp. 81-92, Jan. 2015.
<https://doi.org/10.1016/j.cpc.2014.09.011>
- [32] N. Wagner, *Theory and Applications of Colloidal Suspension Rheology*. Cambridge University Press, 2021.
- [33] M. A. Pindrus, J. L. Cole, J. Kaur, S. J. Shire, S. Yadav, D. S. Kalonia, “Effect of Aggregation on the Hydrodynamic Properties of Bovine Serum Albumin,” *Pharmaceutical Research*, vol. 34, no. 11, pp. 2250-2259, Nov. 2017. <https://doi.org/10.1007/s11095-017-2231-2>