

# Implementing classical transforms for audio signal processing on ESP32

Santos Santander Joaquín Alejandro<sup>1</sup>; Arguelles Bendezú Irene<sup>2</sup>



<sup>1,2</sup>Universidad Tecnológica del Perú, Perú, [U22214608@utp.edu.pe](mailto:U22214608@utp.edu.pe), [C25830@utp.edu.pe](mailto:C25830@utp.edu.pe)

## *Abstract:*

*This study addresses the practical implementation of classical digital signal processing techniques, specifically the Fast Fourier Transform (FFT) and the Z-transform, on the ESP32 microcontroller, an inexpensive and versatile platform programmed using the Arduino IDE. Spectral analysis and audio filtering examples were developed, applying the FFT to identify frequency components and suppress unwanted ranges through spectral editing, and using the Z-transform to design a band-stop FIR filter. Experimental validation was carried out with a PCM5102A DAC, with the outputs observed on an oscilloscope, which demonstrated real-time signal reconstruction and manipulation. The results showed that the FIR filter based on the Z-transform exhibited lower latency and greater temporal continuity compared to the FFT method, making it more suitable for embedded applications. Overall, this work constitutes an accessible educational resource that bridges mathematical theory and experimental practice, fostering learning and innovation in digital audio processing on low-cost platforms.*

*Keywords: audio processing, FFT, Z-transform, digital signals, ESP32, embedded systems.*

# Implementación de transformaciones clásicas para el procesamiento de señales de audio en ESP32

Santos Santander Joaquín Alejandro<sup>1</sup>; Arguelles Bendezú Irene<sup>2</sup>

<sup>1,2</sup>Universidad Tecnológica del Perú, Perú, [U22214608@utp.edu.pe](mailto:U22214608@utp.edu.pe), [C25830@utp.edu.pe](mailto:C25830@utp.edu.pe)

**Resumen:** El presente estudio aborda la implementación práctica de técnicas clásicas de procesamiento digital de señales, específicamente la Transformada Rápida de Fourier (FFT) y la transformada Z, en el microcontrolador ESP32, un hardware económico y versátil programado mediante Arduino IDE. Se desarrollaron ejemplos de análisis espectral y filtrado de audio, aplicando la FFT para identificar componentes frecuenciales y suprimir rangos no deseados mediante edición espectral, y utilizando la transformada Z para diseñar un filtro FIR rechaza-banda. La validación experimental se realizó empleando un DAC PCM5102A y observando las salidas en un osciloscopio, lo que permitió evidenciar la reconstrucción y manipulación de señales en tiempo real. Los resultados mostraron que el filtro FIR basado en la transformada Z presentó menor latencia y mayor continuidad temporal en comparación con el método mediante FFT, lo que lo hace más robusto para aplicaciones embebidas. En conjunto, este trabajo constituye un recurso didáctico que conecta teoría matemática y práctica experimental, fomentando el aprendizaje y la innovación en el procesamiento digital de audio en plataformas de bajo costo.

**Palabras clave:** Procesamiento de audio, FFT, transformada Z, señales digitales, ESP32, sistemas embebidos.

## I. INTRODUCCIÓN

El procesamiento de señales de audio es una disciplina fundamental en el desarrollo de dispositivos digitales, con aplicaciones clave en sistemas de grabación, transmisión y reproducción de sonido [1], [2]. En este contexto, las herramientas matemáticas clásicas son esenciales para modelar, analizar y manipular señales de forma eficiente tanto en el dominio del tiempo como en el de la frecuencia. Estas técnicas permiten abordar desafíos como la reducción de ruido, el filtrado y la síntesis de audio, optimizando así la calidad y el rendimiento de los sistemas embebidos.

Este artículo se centra en la aplicación de varios algoritmos matemáticos bien establecidos utilizados en el procesamiento digital de audio, incluyendo el teorema de Nyquist, las series de Fourier, la transformada de Fourier y la transformada Z [3], [4], [5], [6], [7].

El teorema de Nyquist establece que la frecuencia de muestreo debe ser al menos el doble de la mayor frecuencia presente en la señal para evitar el aliasing, fenómeno que puede distorsionar la reconstrucción de la señal [2]. Las series de Fourier permiten descomponer funciones periódicas en componentes sinusoidales [3], mientras que la **transformada de Fourier** amplía este enfoque para analizar señales en el dominio de la frecuencia, ofreciendo herramientas fundamentales para el análisis espectral y el filtrado [4]. Aunque la transformada de Laplace se utiliza

tradicionalmente en el análisis de sistemas en tiempo continuo [5], en este trabajo solo se revisa desde un enfoque teórico. Por su parte, la transformada Z, contraparte discreta de la de Laplace, es fundamental en el procesamiento digital de señales en microcontroladores y procesadores digitales de señales (DSPs) [5], [6].

La metodología combina una revisión teórica con la implementación práctica en la plataforma ESP32, un microcontrolador económico y versátil, programado mediante Arduino IDE. Las pruebas se realizaron tanto con señales sintéticas como con audio real, aplicando la FFT para el análisis y filtrado espectral, y la transformada Z para el diseño de filtros FIR.

Este enfoque es especialmente relevante en contextos educativos y de prototipado, donde comprender la relación entre teoría matemática y procesamiento de señales en hardware de recursos limitados es crucial. La contribución principal de este trabajo radica en demostrar que técnicas clásicas como la FFT y la transformada Z pueden aplicarse efectivamente en un ESP32 para realizar filtrado en frecuencia, validando experimentalmente en hardware real mediante el DAC PCM5102A y observación en osciloscopio.

## II. TEORÍA

### a) Señales analógicas y digitales:

Para comprender correctamente el análisis de audio en dispositivos digitales, es esencial conocer el comportamiento de las señales analógicas y digitales [8] utilizan ampliamente en telecomunicaciones, presentan diferencias fundamentales. Aunque ambas señales analógicas pueden tomar un rango continuo de valores, como variaciones de voltaje, corriente u otras magnitudes físicas.

En contraste, las señales digitales oscilan entre valores discretos representados como bajo (0) y alto (1). En la figura 1 se observa el comportamiento característico de cada una[9].

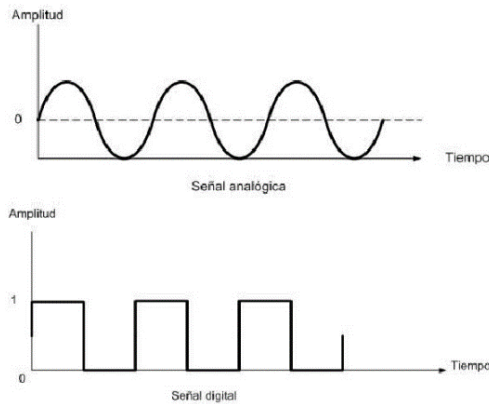


Figura 1: Señales analógicas y digitales  
Fuente: [9].

#### b) Conversor Analógico-Digital:

Una vez repasados los conceptos de las señales analógicas y digitales es fundamental entender cómo un dispositivo digital procesar señales analógicas para su correcta reconstrucción. Esto se logra mediante un Conversor Analógico-Digital (ADC) [10]. Este dispositivo convierte señales analógicas a datos digitales discretos a través de un proceso de muestreo y cuantización. De esta manera un microcontrolador o un procesador de señales digitales (DSP) pueden interpretar y procesar señales de entrada, permitiendo aplicaciones como el análisis, la grabación y la transmisión de audio.

#### c) Protocolo I2S:

El Inter-IC Sound (I2S) es un protocolo de comunicación síncrono diseñado específicamente para la transmisión de datos de audio digital entre circuitos integrados. Requiere de una señal de reloj de bit (**BCLK** o **LCK**) que sincroniza la transferencia de cada bit, una señal de selección de palabra (**WS**) que indica si los datos corresponden al canal izquierdo o derecho, y una línea de datos (**SD**) por donde se transmiten las muestras de audio. En algunos casos se emplea además una señal de reloj maestro (**MCLK**) para mejorar la estabilidad. El protocolo I2S es ampliamente utilizado en microcontroladores como el ESP32 para la conexión con códecs de audio, ADCs y DACs externos [11].

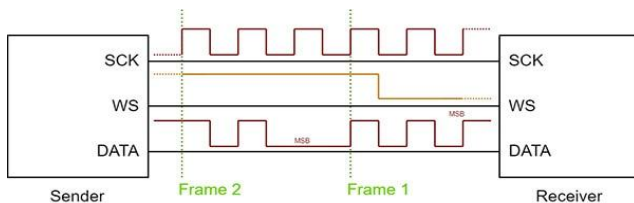


Figura 2: Protocolo I2S.  
Fuente: [11].

#### d) A2DP:

El Advanced Audio Distribution Profile (A2DP) es un perfil del estándar Bluetooth diseñado para la transmisión

inalámbrica de audio de alta calidad en tiempo real. Este protocolo permite enviar flujos estéreo codificados, generalmente en formato SBC o AAC, desde un dispositivo fuente (ej. un microcontrolador o teléfono) hacia un receptor de audio (ej. un altavoz o DAC externo)[12].

#### e) Teorema de Nyquist:

El teorema de Nyquist resulta muy conveniente al momento de reconstruir una señal continua con lecturas discretas como aquellas que proporciona el ADC, este teorema explica que, para una correcta recreación de una señal, es necesario que la frecuencia de muestreo del ADC sea al menos el doble de la frecuencia máxima de la señal tal y cómo se muestra en la ecuación 1 y 2. Esto asegura que no ocurra aliasing, un fenómeno que introduce errores al capturar señales de frecuencias más altas como si fueran más bajas[13].

$$f_{\text{muestreo}} = 2 \cdot f_{\text{máxima}} \quad (1)$$

En el contexto de este artículo se estudian los dispositivos de audio, teniendo en cuenta que la frecuencia de audio varía entre 20 Hz y 20 kHz, por lo que tenemos ecuación 2.

$$f_{\text{muestreo}} = 2 \cdot 20 \text{ kHz} = 40 \text{ kHz} \quad (2)$$

Observando los resultados, se concluye que se requiere de una frecuencia de 40 kHz para la reconstrucción de una señal de audio.

#### f) Serie de Fourier:

Damián Pedraza define a la serie de Fourier de la siguiente manera “No es otra cosa que una suma de senos y cosenos de frecuencias determinadas y amplitudes determinadas” [14], en otras palabras, es una función periódica que puede representarse como una sumatoria infinita de senos y cosenos, cuya amplitud y frecuencia varían. La fórmula base de la serie de Fourier es la ecuación 3.

$$F(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left[ a_n \cos\left(\frac{2\pi \cdot n \cdot t}{T}\right) + b_n \sin\left(\frac{2\pi \cdot n \cdot t}{T}\right) \right] \quad (3)$$

Para calcular las constantes  $a_0$ ,  $a_n$  y  $b_n$  se emplean las ecuaciones 4, 5 y 6 respectivamente [15].

$$a_0 = \frac{2}{T} \int_{-T/2}^{T/2} F(t) dt \quad (4)$$

$$a_n = \frac{2}{T} \int_{-T/2}^{T/2} \cos\left(\frac{2\pi \cdot n \cdot t}{T}\right) * F(t) dt \quad (5)$$

$$b_n = \frac{2}{T} \int_{-T/2}^{T/2} \sin\left(\frac{2\pi * n * t}{T}\right) * F(t) dt \quad (6)$$

Debido a la propiedad del exponente complejo, la serie de Fourier también se puede representar con la ecuación 7 [16].

$$F(t) = \sum_{n=-\infty}^{\infty} C_n e^{-\frac{i2\pi nt}{T}} \quad (7)$$

Para calcular la constante  $C_n$  se emplea la ecuación 8.

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} F(t) e^{-\frac{i2\pi nt}{T}} dt \quad (8)$$

#### g) Transformada de Fourier:

La transformada de Fourier es una herramienta matemática fundamental que permite descomponer la señal en sus componentes armónicos [17]. Esta descomposición facilita el análisis de señales y sistemas en aplicaciones como el procesamiento de señales, comunicaciones y reducción de ruido. La fórmula general de la transformada de Fourier es la 9.

$$C_n = \frac{1}{T} \int_{-T/2}^{T/2} F(t) e^{-\frac{i2\pi nt}{T}} dt \quad (9)$$

Donde:

$f(t)$ : Es la señal en el dominio del tiempo.

$t$ : Representa el tiempo.

$F(\omega)$ : Es la señal transformada en el dominio de la frecuencia.

Nota adicional:

El resultado  $F(\omega)$  es una función compleja que proporciona tanto la amplitud como la fase de cada componente de frecuencia de la señal original.

#### h) Transformada discreta de Fourier:

En sistemas digitales, donde las señales son discretas y tienen una longitud finita, se emplea la transformada discreta de Fourier (DFT). Esta transformada convierte una señal discreta  $x[n]$  en su equivalente en el dominio de la frecuencia, adaptándose a las limitaciones y características del procesamiento digital [18]. Su implementación se logra mediante la fórmula 10.

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}} \quad (10)$$

Donde:

$x[n]$ : Es la señal en el dominio del tiempo discreto.

$N$ : Es el número total de muestras.

$k$ : Es el índice de frecuencia, que toma valores entre 0 y  $N-1$ .

$X[k]$ : Representa la señal transformada en el dominio de la frecuencia discreta.

Nota adicional:

La DFT produce una representación discreta y finita de las frecuencias de la señal, con un intervalo de frecuencias limitado a  $[0, \text{frecuencia de Nyquist}]$ . La periodicidad en el dominio de la frecuencia es una consecuencia del muestreo discreto.

#### i) Transformada rápida de Fourier:

La transformada de rápida de Fourier (FFT) es un algoritmo eficiente para calcular la DFT, logrando una reducción significativa en la complejidad computacional. Mientras que la DFT tiene una complejidad de  $O(N^2)$ , la FFT utiliza propiedades como la periodicidad y simetría de las exponenciales complejas  $O(N \log N)$  [19].

Ventajas de la FFT:

- Reduce drásticamente el tiempo de procesamiento, especialmente para señales con un gran número de muestras.
- Es ampliamente utilizada en procesamiento de audio, análisis de imágenes, telecomunicaciones y sistemas de radar.

Aunque la FFT y la DFT comparten la misma fórmula, la FFT divide el problema en subproblemas más pequeños mediante un enfoque recursivo, permitiendo un cálculo mucho más rápido. Un ejemplo destacado de implementación es el algoritmo de Cooley-Tukey.

#### j) Transformada inversa de Fourier:

La transformada inversa de Fourier (IFT) permite reconstruir la señal original en el dominio del tiempo a partir de su representación en el dominio de la Frecuencia [20]. Es la herramienta complementaria a la transformada de Fourier y se utiliza para regresar al dominio temporal después del análisis en frecuencias. Su fórmula general es la 11.

$$f(t) = \int_{-\infty}^{\infty} F(\omega) e^{i2\pi\omega t} d\omega \quad (11)$$

Donde:

$f(t)$ : Es la señal reconstruida en el dominio del tiempo.

$\omega$ : Representa la frecuencia angular.

$F(\omega)$ : Es la señal transformada en el dominio de la frecuencia.

#### k) Transformada discreta inversa de Fourier:

En el procesamiento digital, la transformada inversa discreta de Fourier (IDFT) se emplea para recuperar la señal

original en el dominio temporal discreto a partir de su representación en el dominio de la frecuencia discreta. Su implementación se logra mediante la fórmula 12.

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{\frac{j2\pi kn}{N}} \quad (12)$$

Donde:

$x[n]$ : Es la señal en el dominio del tiempo discreto.

$N$ : Es el número total de muestras.

$X[k]$ : Son los valores en el dominio de la frecuencia discreta.

$k$ : Es el índice que representa las distintas frecuencias discretas.

#### l) Transformada rápida inversa de Fourier:

La transformada rápida inversa de Fourier (IFFT) es una implementación optimizada de la IDFT que permite calcular la señal original en el dominio temporal de manera mucho más eficiente. Al igual que la FFT, la IFFT reduce la complejidad computacional de  $O(N^2)$  a  $O(N \log N)$ , haciéndola ideal para aplicaciones que requieren cálculos rápidos.

Ventajas de la IFFT:

- Reduce drásticamente el tiempo de procesamiento, especialmente grandes conjuntos de datos.
- Se utiliza en sistemas de telecomunicaciones (como OFDM), procesamiento de audio y análisis espectral, donde la reconstrucción eficiente de la señal es crucial.

Aunque la FFT y la DFT comparten la misma fórmula, la FFT divide el problema en subproblemas más pequeños mediante un enfoque recursivo, permitiendo un cálculo mucho más rápido.

#### m) Transformada de Laplace:

La transformada de Laplace es una herramienta matemática que permite convertir una función del dominio temporal  $f(t)$  en una representación en el dominio complejo  $F(s)$  [21]. Este proceso es ampliamente utilizado en el procesamiento de señales, así como en la resolución de ecuaciones diferenciales y sistemas lineales, debido a que simplifica el análisis matemático. Su implementación se logra mediante la fórmula 13.

$$F(s) = \int_0^{\infty} f(t) e^{-st} dt \quad (13)$$

Donde:

$F(s)$ : Es la función representada en el dominio complejo.

$s$ : Es la variable compleja con componente real e imaginario ( $s = \sigma + j\omega$ ).

$f(t)$ : Es la función en el dominio del tiempo.

$t$ : Es el dominio del tiempo.

#### n) Transformada inversa de Laplace:

La transformada inversa de Laplace permite convertir una función representada en el dominio complejo  $F(s)$  al dominio temporal  $f(t)$ . Este procedimiento es fundamental para recuperar la señal original después de su procesamiento [22], y es ampliamente utilizado en el análisis de señales y resolución de sistemas dinámicos. Su implementación se logra mediante la fórmula 14.

$$f(t) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F(s) e^{st} ds \quad (14)$$

Donde:

$F(s)$ : Es la función representada en el dominio complejo.

$s$ : Es la variable compleja con componente real e imaginario ( $s = \sigma + j\omega$ ).

$f(t)$ : Es la función en el dominio del tiempo.

$t$ : Es el dominio del tiempo.

$c$ : Es un valor real que asegura que la línea de integración esté a la derecha de todas las singularidades de  $F(s)$ .

#### o) Transformada Z:

La transformada Z es una herramienta matemática para analizar y procesar señales discretas, especialmente en dispositivos digitales como microcontroladores o procesadores de señales digitales (DSP). A diferencia de la transformada de Laplace, que se aplica en sistemas continuos, la transformada Z se adapta a señales definidas en instantes discretos de tiempo, lo que la hace fundamental en el análisis y diseño de sistemas de control digital. Su implementación se logra mediante la fórmula 15.

$$X(z) = \sum_{n=0}^{N-1} x[n] z^{-n} \quad (15)$$

Donde:

$X(z)$ : Es la representación de la señal  $x[n]$  en el dominio Z.

$x[n]$ : Es la señal discreta en el dominio temporal.

$z$ : Es la variable compleja en el dominio Z, definida como  $z = re^{j\omega}$ , donde  $r$  es el módulo y  $\omega$  es la frecuencia angular.

$n$ : Es el índice de tiempo discreto.

#### p) Transformada Z inversa:

La transformada Z inversa permite recuperar la señal original  $x[n]$  en el dominio temporal a partir del dominio complejo Z  $X(z)$ . Este proceso es esencial en sistemas digitales para analizar y sintetizar señales digitales [23]. Su implementación se logra de con la ecuación 16.



$$x[n] = \frac{1}{2\pi i} \oint X(z) z^{n-1} dz \quad (16)$$

Donde:

$X(z)$ : Es la representación de la señal  $x[n]$  en el dominio  $Z$ .

$x[n]$ : Es la señal discreta en el dominio temporal.

$z$ : Es la variable compleja en el dominio  $Z$ , definida como  $z=re^{j\omega}$ , donde  $r$  es el módulo y  $\omega$  es la frecuencia angular.

$n$ : Es el índice de tiempo discreto.

#### q) Filtro FIR

Los filtros de respuesta finita al impulso (Finite Impulse Response) tienen como objetivo modificar selectivamente el contenido espectral de una señal de audio, atenuando un rango de frecuencias y preservando otras [24]. Su diseño se fundamenta en la **transformada  $Z$** , ya que esta permite expresar la función de transferencia como un polinomio en potencias negativas de  $z$ , lo cual simplifica el cálculo de la salida al representar la convolución directa entre la señal de entrada y los coeficientes del filtro (*taps*).

La ecuación en tiempo discreto que define a un filtro FIR de orden  $N-1$  es la fórmula 17.

$$y[n] = \sum_{i=0}^{N-1} h[i] x[n-i] \quad (17)$$

Donde:

$n$ : posición actual de la entrada.

$N$ : número de muestras.

$x[n]$ : entrada de la señal.

$y[n]$ : salida de la señal.

$h[n]$ : coeficientes del filtro.

Para obtener los coeficientes de un filtro FIR ideal de tipo pasa-bajos se utiliza la función *sinc*, que resulta de truncar la respuesta impulsional infinita de un filtro ideal. Su expresión es la fórmula 18.

$$h[n] = \frac{2f_c}{f_s} \text{sinc} \left( \frac{2\pi f_c (n - \frac{N-1}{2})}{f_s} \right) \quad (18)$$

Donde:

$n$ : posición actual de la entrada.

$N$ : número de muestras.

$f_c$ : frecuencia de corte.

$f_s$ : frecuencia de muestreo.

$h[n]$ : coeficientes de la transformada.

La secuencia ideal  $h[n]$  obtenida mediante la función *sinc* se truncó a  $N$  coeficientes y se multiplicó por una ventana de Hamming, con el fin de reducir la fuga espectral y mejorar la atenuación en la banda de rechazo. Este procedimiento permitió implementar un filtro FIR de tipo rechaza-banda (notch) estable y de fase lineal.

#### r) Conversor digital-analógico:

El Conversor Digital-Analógico (DAC) es un dispositivo que transforma una señal digital procesada en una señal analógica continua, permitiendo su reproducción en dispositivos como amplificadores o altavoces. Este proceso implica una transformación inicial de valores digitales a niveles discretos de valores de voltaje o de corriente, seguida de una reconstrucción de la señal mediante técnicas de interpolación y el uso de filtros pasa bajo para suavizar la salida. Los DAC son esenciales en sistemas de audio, telecomunicaciones y dispositivos médicos, ya que garantizan una reproducción fiel y eficiente de las señales procesadas, dependiendo de parámetros como la resolución y la frecuencia de muestreo [25].

### III. METODOLOGIA

La metodología consistió en implementar y validar transformadas clásicas en el entorno Arduino IDE sobre un ESP32. Como señal de entrada se utilizó un flujo de audio real transmitido mediante A2DP. Se plantearon dos pruebas principales:

- Procesamiento con FFT para obtener el espectro y aplicar un filtrado por supresión de componentes específicas (FFT de 1024 puntos, frecuencia de muestreo de 44.1 kHz).
- Procesamiento con transformada  $Z$ , diseñando un filtro FIR de orden 63 mediante la función *sinc* y ventana de Hamming para suprimir un rango de frecuencias determinado.

En ambos casos, las señales se analizaron antes y después de aplicar las transformadas, y se reconstruyeron con un DAC PCM5102A mediante protocolo I2S. Los resultados se validaron comparando representaciones gráficas (FFT en el plotter serial) y observaciones experimentales en un osciloscopio.

En cada implementación se analizaron las señales antes y después de aplicar la transformada correspondiente. Para validar los resultados se implementó un filtro que suprimió un rango de frecuencias predefinido y posteriormente se reconstruyó la señal. La conversión de digital a analógico se realizó con un DAC PCM5102A mediante el protocolo I2S, cuya salida fue observada en un osciloscopio para verificar el desempeño experimental.

Cabe resaltar que la transformada de Laplace no fue considerada en esta etapa, dado que su aplicación está restringida a sistemas continuos, mientras que la transformada  $Z$  se adapta de manera más natural a los sistemas discretos propios de dispositivos digitales.

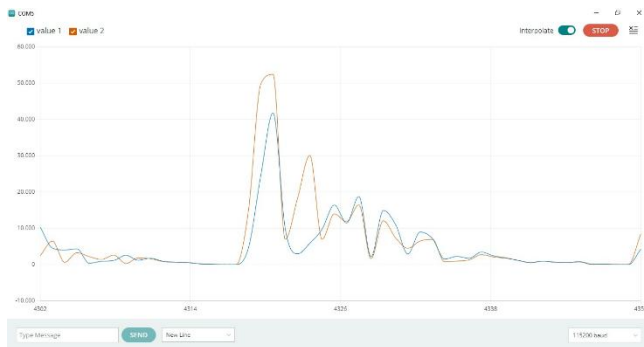


Figura 3: Transformada rápida de Fourier aplicada en una señal de audio.  
Fuente: Propia.

#### a. IMPLEMENTACIÓN DE LA TRANSFORMADA RÁPIDA DE FOURIER:

En esta etapa se diseñó un filtro digital aplicando la transformada rápida de Fourier (FFT). La entrada de la señal correspondió a un flujo de audio transmitido mediante A2DP, muestrea a 44100hz debido a que cumple con el teorema de Nyquist, además, se usaron bloques de 1024 muestras. La figura 3 muestra la representación espectral obtenida en el Serial Plotter del entorno Arduino IDE, donde se aprecian los componentes frecuenciales de la señal de audio. Este tipo de análisis permite caracterizar con precisión el contenido espectral, base de aplicaciones como asistentes de voz, sistemas de reconocimiento y analizadores de espectro.

El procedimiento consistió en implementar un **filtro digital de tipo rechaza-banda** sobre la representación en frecuencia, eliminando componentes entre **85 Hz y 3 kHz**. Posteriormente, se aplicó la transformada inversa (IFFT) para reconstruir la señal en el dominio temporal. Con el fin de validar el desempeño, la señal reconstruida fue enviada a través del DAC **PCM5102A** y observada en un osciloscopio. La figura 4 corresponde a la señal original, mientras que la figura 5 muestra la misma señal tras la aplicación del filtro, donde se aprecia la atenuación de las componentes eliminadas.



Figura 4: señal de audio de la FFT antes del filtro.  
Fuente: Propia.

#### b. IMPLEMENTACIÓN DEL FILTRO FIR:

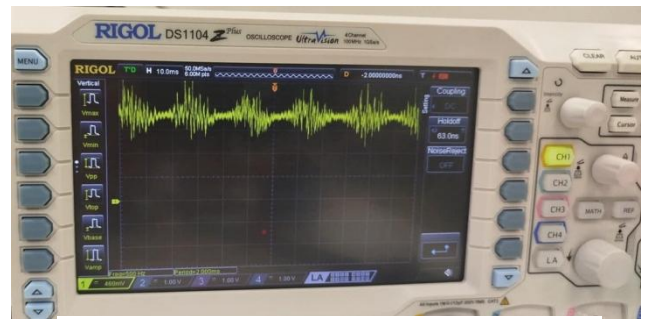


Figura 5: señal de audio de la FFT después del filtro.  
Fuente: Propia.

Tras la prueba realizada con la FFT, se implementó un **filtro digital mediante la transformada Z**, diseñado como un **filtro FIR de 63 coeficientes** con frecuencia de muestreo de **44.1 kHz**. El objetivo fue replicar el mismo comportamiento del filtro aplicado en el dominio frecuencial, es decir, un filtro de tipo **rechaza-banda (notch)**.

La figura 6 muestra la señal de audio antes de aplicar el filtrado, mientras que la figura 7 presenta la misma señal tras la aplicación del FIR. En esta última se observa la atenuación de las componentes frecuenciales eliminadas, validando así la eficacia del diseño basado en la transformada Z.

### IV. RESULTADOS:

En esta sección se presentan y analizan los resultados adquiridos al aplicar la transformada rápida de Fourier (FFT) y la transformada Z mediante un filtro de respuesta finita al impulso (FIR) rechaza-banda (notch) sobre señales de audio procesadas en la placa ESP32. Se incluyen las capturas de la metodología, ya sea sobre el dominio espectral o sobre el dominio temporal, para validar la implementación en un software real.

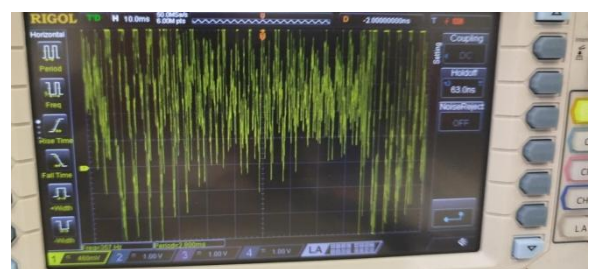


Figura 6: señal de audio del filtro FIR antes del filtro.  
Fuente: Propia.

#### a. IMPLEMENTACIÓN DE LA TRANSFORMADA RÁPIDA DE FOURIER:

En este apartado se realizó un filtro rechaza-banda utilizando la FFT, para realizarlo se utilizó una frecuencia de muestreo de 44,1khz junto con bloques de 1024 muestras, el protocolo Bluetooth A2DP proporciona la señal de audio, una vez recibido dicha señal, será sometida a la

FFT para analizar sus componentes espectrales como se aprecia en la figura 3.

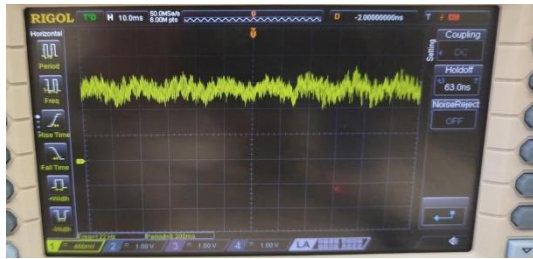


Figura 7: señal de audio del filtro FIR después del filtro.  
Fuente: Propia.

Una vez realizada dicha transformada, se elige desde un botón si eliminar o no los componentes espectrales dentro del rango de entre 85hz a 3khz. A continuación, realizar la transformada rápida inversa de Fourier (IFFT) para restaurar la señal al dominio temporal.

En la figura 4 se aprecia una señal de audio sometida a la FFT y a la IFFT sin aplicar el filtro. Mientras que en la figura 5 se aplicó el filtro

#### b. IMPLEMENTACIÓN DE LA TRANSFORMADA Z:

Para la implementación de la transformada Z se realizó un filtro FIR, el cual requiere de la misma frecuencia de muestreo aplicada en la FFT, a diferencia de que esta vez se usan bloques de 63 muestras. El filtro FIR es rechaza-banda de entre 85hz a 3khz, primero se calcularon los coeficientes  $h[n]$  para después utilizarlos en el filtro.

La placa ESP32 espera a recibir una señal de audio desde el mismo protocolo propuesto anteriormente. Una vez captada la señal se elige mediante un botón si aplicar el filtro o no.

En la figura 6 se aprecia la señal de audio sin aplicar el filtro FIR. Mientras que la figura 7 muestra la misma señal aplicando el filtro.

#### c. IMPLEMENTACIÓN DE LA TRANSFORMADA RÁPIDA INVERSA DE FOURIER:

Después de realizar la FFT, se aplicó un filtro para aislar el componente de 10kHz de la señal. Posteriormente, se utilizó la IFFT para restaurar la señal, visualizándose el resultado en la Figura 4. Este procedimiento evidencia la eficacia combinada de la FFT y la IFFT en aplicaciones prácticas de procesamiento de señales digitales de audio, destacando su utilidad para tareas de análisis y síntesis.

#### d. Comparación entre el filtro de FFT y el filtro FIR:

En las figuras 4 y 5 se aprecia el efecto del código de la FFT en la salida de la señal, mientras que las figuras 6 y 7 muestran el efecto del filtro FIR.

- Al analizar la señal en el filtro FFT se aprecian interrupciones periódicas en la señal, esto es

debido a que el filtro requiere 17 milisegundos para aplicarse en su totalidad, haciéndolo poco eficaz ya que dichas interrupciones son perceptibles desde el oído humano.

- Analizando el filtro FIR se aprecia que la señal de audio carece de interrupciones, a diferencia del filtro FFT, el filtro FIR solo requiere de 6 milisegundos para aplicarse en su totalidad, haciéndolo mucho más eficaz que el filtro FFT.

#### e. DISCUSION DE LOS RESULTADOS:

Analizando el comportamiento de ambos filtros se llega a la conclusión que el filtro FIR es mucho más eficaz que el filtro FFT. Sin embargo, esto no hace que la FFT sea inútil para el procesamiento digital de audio, solo que la FFT no está pensada para realizar filtros, solo para sacar información relevante de una señal. Es decir, para identificar el tipo de señal que se está recibiendo, la FFT gana relevancia, mientras que un filtro FIR solo elimina frecuencias de una señal. En asistente de IA mediante voz, ambos conceptos son altamente relevantes, el filtro FIR elimina frecuencias parasitas de las frecuencias requeridas, mientras que la FFT analiza la salida del filtro para sacar la información relevante de la señal para su posterior análisis.

En términos de latencia, el enfoque basado en FFT requiere la adquisición de bloques de 1024 muestras, lo que introduce un retardo cercano a 17 ms a 44.1 kHz, además del tiempo de cómputo. En contraste, el filtro FIR de 63 coeficientes introduce un retardo de grupo aproximado de 0.7 ms, lo que lo hace más adecuado para aplicaciones de audio en tiempo real. Asimismo, la continuidad temporal de la señal es mayor en el FIR, ya que procesa muestra a muestra sin necesidad de segmentación, mientras que la FFT opera en bloques discretos que pueden generar discontinuidades si no se emplean técnicas de solapamiento. Finalmente, en cuanto a robustez, los FIR ofrecen estabilidad garantizada y fase lineal, mientras que el método espectral con FFT es más flexible para editar componentes específicas, aunque su desempeño depende fuertemente de parámetros como el tamaño de la ventana y el solapamiento aplicado.

Si bien este trabajo se centra en la aplicación de transformadas clásicas de procesamiento digital de señales (FFT y Z) sobre audio en tiempo real, resulta interesante señalar que los principios de Fourier se han expandido hacia dominios más complejos, como el procesamiento de señales en grafos. En [4] se introduce la *Graph Fourier Transform (GFT)*, que permite analizar señales en redes y estructuras irregulares, mostrando la vigencia y versatilidad de las ideas de Fourier incluso en el contexto de la ciencia de datos moderna.



## V. CONCLUSIONES:

En este trabajo se abordó la implementación de transformadas clásicas de procesamiento digital de señales en un microcontrolador ESP32, demostrando su aplicabilidad en el análisis y filtrado de audio. La transformada de Fourier permitió obtener la representación espectral de las señales y aplicar filtrado por edición de componentes en el dominio de la frecuencia mediante FFT e IFFT, mientras que la transformada Z facilitó el diseño de un filtro FIR de tipo rechaza-banda con 63 coeficientes, ejecutado directamente en el dominio temporal.

Los resultados experimentales, validados mediante capturas en el *Serial Plotter* y en un osciloscopio conectado a la salida de un DAC PCM5102A, confirmaron que ambos enfoques permiten la supresión efectiva de componentes indeseadas. En particular, el filtro FIR basado en la transformada Z presentó menor latencia (6 ms frente a  $\approx 17$  ms en el método FFT) y mayor continuidad temporal en comparación con el método de la FFT, lo cual lo hace más robusto para aplicaciones en tiempo real.

Más allá de su valor práctico, la propuesta constituye un recurso didáctico accesible para la enseñanza de técnicas de DSP en hardware económico, permitiendo a estudiantes y desarrolladores explorar conceptos como análisis espectral, filtrado digital y reconstrucción de señales en un entorno experimental sencillo.

## VI. LIMITACIONES Y TRABAJO FUTURO:

No obstante, el estudio presenta limitaciones, ya que los resultados obtenidos fueron principalmente cualitativos, sin incluir métricas detalladas de consumo de CPU, memoria o latencia. Como trabajo futuro, se plantea la integración de pruebas cuantitativas, la liberación del código en repositorios abiertos para garantizar reproducibilidad, y la extensión de la metodología a escenarios más complejos, como reconocimiento de voz o procesamiento de audio en condiciones reales.

## REFERENCIAS:

- [1] A. V. Zaytsev, "Digital Signals Processing," in *Lecture Notes in Networks and Systems*, 2022. doi: 10.1007/978-3-030-94873-3\_31.
- [2] Baraniuk Richard, "10.5: Fenómenos de Aliasing - LibreTexts Español." Accessed: Jan. 24, 2025. [Online]. Available: [https://espanol.libretexts.org/Ingenieria/Se%C3%B1ales\\_y\\_Sistemas\\_\(Baraniuk\\_et\\_al.\)/10%3A\\_Muestreo\\_y\\_Reconstrucci%C3%B3n/10.05%3A\\_Fen%C3%B3menos\\_de\\_Aliasing](https://espanol.libretexts.org/Ingenieria/Se%C3%B1ales_y_Sistemas_(Baraniuk_et_al.)/10%3A_Muestreo_y_Reconstrucci%C3%B3n/10.05%3A_Fen%C3%B3menos_de_Aliasing)
- [3] Camarasa Miguel, "(19) ¿Qué es una Serie de Fourier? - YouTube." Accessed: Jan. 24, 2025. [Online]. Available: <https://www.youtube.com/watch?v=Mdk6BWeVNIs>
- [4] B. Ricaud, P. Borgnat, N. Tremblay, P. Gonçalves, and P. Vanderghenst, "Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs," 2019. doi: 10.1016/j.crhy.2019.08.003.
- [5] L. Obando, "La Transformada Z – Análisis de sistemas discretos," 2020.
- [6] Schneider Josh and Smalley Ian, "¿Qué es un microcontrolador? | IBM." Accessed: Jan. 24, 2025. [Online]. Available: <https://www.ibm.com/mx-es/think/topics/microcontroller>
- [7] Marley Scott, "ESP32 spectrum analyser VU meter using arduinoFFT and a FastLED matrix - YouTube." Accessed: Jan. 09, 2025. [Online]. Available: [https://www.youtube.com/watch?v=Mgh2WbLO5\\_c](https://www.youtube.com/watch?v=Mgh2WbLO5_c)
- [8] Paula Maria, Leslie Antonio, Reyero Raul, Fernández Sergio, and Crespo Rafael, "Diferencias entre señal analógica y digital | VIU Internacional." Accessed: Jan. 15, 2025. [Online]. Available: <https://www.universidadviu.com/int/actualidad/nuestros-expertos/diferencias-entre-senal-analogica-y-digital>
- [9] Rosales César, "Señal analógica y Señal digital'. | Download Scientific Diagram." Accessed: Jan. 15, 2025. [Online]. Available: [https://www.researchgate.net/figure/Senal-analogica-y-Senal-digital\\_fig1\\_362489895](https://www.researchgate.net/figure/Senal-analogica-y-Senal-digital_fig1_362489895)
- [10] ARROW company, "Convertidores analógicos a digitales: ¿cómo funciona un ADC? | Arrow.com." Accessed: Jan. 15, 2025. [Online]. Available: <https://www.arrow.com/es-mx/research-and-events/articles/engineering-resource-basics-of-analog-to-digital-converters>
- [11] Wright Gavin, "¿Qué es el teorema de Nyquist?" Accessed: Jan. 13, 2025. [Online]. Available: <https://www.techtarget.com/whatis/definition/Nyquist-Theorem>
- [12] Pedraza Damian, "SERIE DE FOURIER: Parte 1: Idea Intuitiva | El Traductor - YouTube." Accessed: Jan. 14, 2025. [Online]. Available: [https://www.youtube.com/watch?v=\\_wKus3Ctk7Q&t=6s](https://www.youtube.com/watch?v=_wKus3Ctk7Q&t=6s)
- [13] Pedraza Damian, "SERIE DE FOURIER: Parte 2: La Interpretación | El Traductor - YouTube." Accessed: Jan. 14, 2025. [Online]. Available: <https://www.youtube.com/watch?v=EFGrElc0FRk&t=1s>
- [14] Pedraza Damian, "SERIE DE FOURIER: Parte 3: Serie de Exponenciales Complejas | El Traductor - YouTube." Accessed: Jan. 14, 2025. [Online]. Available: <https://www.youtube.com/watch?v=F0gLIX-gViU>
- [15] L. Sun, D. You, and D. Qi, "Application of Fourier Transform in Signal Processing," *Signal and information Processing*, vol. 1, no. 1, 2017.

- [16] D. A. Jaramillo Chamba and L. Chuquimarca Jiménez, “Estudio comprensivo de la Transformada de Fourier Discreta para el análisis de señales digitales,” *Revista Científica y Tecnológica UPSE*, vol. 9, no. 1, 2022, doi: 10.26423/rctu.v9i1.664.
- [17] S. I. Mesa and P. P. Riera, “La transformada rápida de Fourier: fundamentos y aplicaciones,” Jun. 2024, Accessed: Jan. 10, 2025. [Online]. Available: <https://digibuo.uniovi.es/dspace/handle/10651/72831>
- [18] S. Lu, E. Abozinadah, and E. Erkec, “Image design and interaction technology based on Fourier inverse transform,” *Applied Mathematics and Nonlinear Sciences*, vol. 7, no. 2, 2022, doi: 10.2478/amns.2021.1.00061.
- [19] G. Patané, “Spectral Laplace Transform of Signals on Arbitrary Domains,” *J Sci Comput*, vol. 96, no. 3, 2023, doi: 10.1007/s10915-023-02274-7.
- [20] T. B. Moraes, “Inverse Laplace Transform for signal analysis of low field nuclear magnetic resonance,” *Quim Nova*, vol. 44, no. 8, 2021, doi: 10.21577/0100-4042.20170751.
- [21] Y. Jia and P. Xu, “Convulsive Blind Source Separation for Communication Signals Based on the Sliding Z-Transform,” *IEEE Access*, vol. 8, 2020, doi: 10.1109/ACCESS.2020.2976700.
- [22] Miguel, “Señales analógicas y digitales. - Control Real Español.” Accessed: Jan. 18, 2025. [Online]. Available: <https://controlreal.com/es/senales-analogicas-y-digitales/>