

# JITTER

CARABALLO CARDENAS ANDRES FELIPE, UNIVERSIDAD MILITAR NUEVA GRANADA  
[est.andres.caraballo@unimilitar.edu.co](mailto:est.andres.caraballo@unimilitar.edu.co)

**Resumen** La práctica de laboratorio tuvo como objetivo utilizar el microcontrolador Raspberry Pi Pico 2W para trabajar con su conversor análogo-digital (ADC). Con este se realizaron mediciones de una señal senoidal en diferentes valores de frecuencia, analizándola tanto en el dominio del tiempo como en el dominio de la frecuencia mediante la Transformada Rápida de Fourier (FFT). La señal fue generada con un generador de funciones y posteriormente se ejecutó el programa suministrado por el docente, el cual producía dos archivos de salida: el primero contenía los datos muestreados en función del tiempo y el segundo los resultados del espectro en frecuencia. Finalmente, ambos conjuntos de datos fueron graficados y analizados en MATLAB.

## I. INTRODUCCIÓN

En esta práctica de laboratorio se trabajó con un código fuente compartido por el docente, el cual, al ejecutarse, generaba dos archivos de salida tal como se mencionó en el resumen. El propósito de la actividad fue observar cómo se comporta una señal senoidal al variar su frecuencia y, además, analizar de qué manera el número de puntos utilizados en la Transformada Rápida de Fourier (FFT) influye en su representación. De esta forma, se buscó comparar las gráficas obtenidas y reconocer las diferencias más relevantes entre ellas.

## II. PROCEDIMIENTO

En esta sección se describe el procedimiento para el desarrollo de cada uno de los puntos propuestos en la guía de laboratorio.

### A. Configuración del generador de señales

En este ítem se configura el generador de señales para producir una onda senoidal con una frecuencia de 200 Hz, una amplitud de 1.2 Vpp y un nivel DC de 1.6 V, como se muestra en la siguiente imagen.

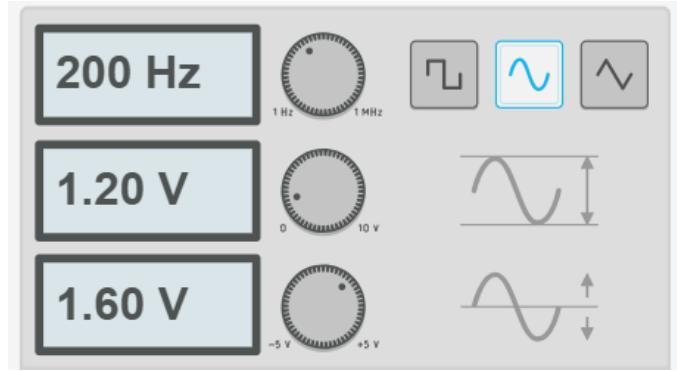


Figura 1. Configuración del generador de señales

Luego de haber configurado el generador se procede a verificar la señal en el osciloscopio la cual se muestra a continuación.

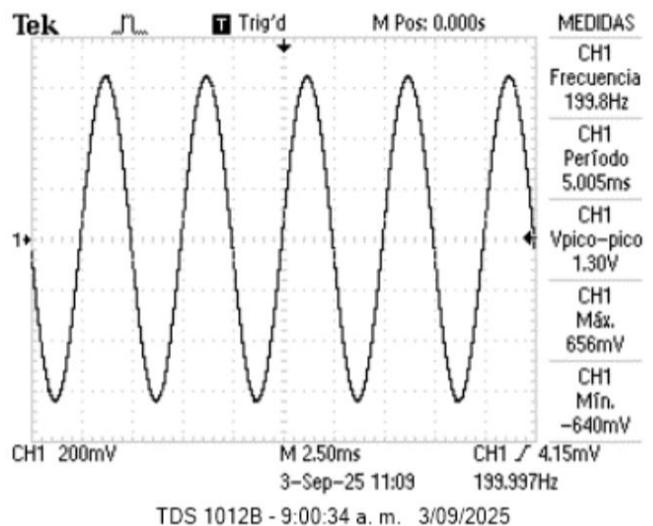


Figura 2. Señal senoidal resultante

Una vez visualizada la señal, se conecta a una de las entradas del conversor análogo-digital (ADC) de la Raspberry Pi Pico 2W, específicamente al pin número 26.

### B. Ejecución del código ADC\_testing.py

En esta sección se carga y ejecuta en el microcontrolador el código compartido por el docente. Al finalizar la compilación, se generan dos archivos: el primero corresponde a las muestras de la señal en el dominio del tiempo y el segundo a las magnitudes en el dominio de la frecuencia. Finalmente, el programa entrega la información mostrada en la siguiente imagen.

```
MEY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.74 Hz
Offset DC removido: 1.474 V
Frecuencia dominante: 199.84 Hz
Amplitud señal: 0.158 V
Piso de ruido: 0.00184 V (-65.10 dB FS)
SNR: 38.67 dB, ENOB: 6.13 bits
```

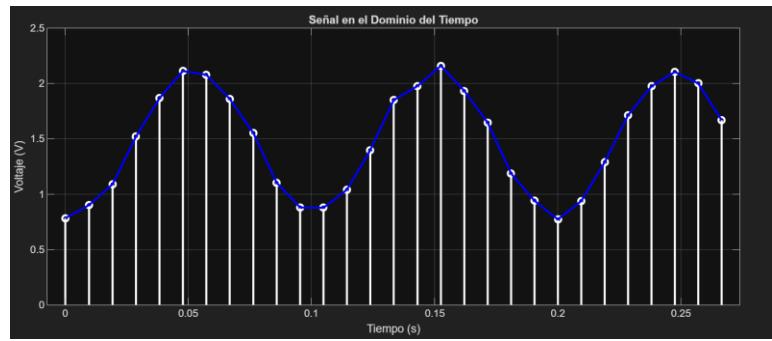
**Figura 3. Información mostrada por el programa**

La explicación que tiene cada línea es:

- **Frecuencia deseada:** Se establece que la frecuencia es 2000 Hz, pero en la práctica se medio a un valor cercano de 1894.74 Hz.
- **Offset DC removido:** Indica un valor 1.474 V el cual hace referencia el nivel de corriente continua presente en la señal la cual fue eliminada.
- **Frecuencia dominante:** Es el valor que indica la frecuencia principal que se detecta en la señal medida en este caso 199.84 Hz.
- **Amplitud de la señal:** Indica el valor de la amplitud la cual es de 0.158 V
- **Piso ruido:** Es el nivel de ruido que está presente en la medición la cual marca un valor en voltaje de 0.00018 V que al ser convertida en dB marca -65.10 lo cual significa que el ruido es muy pequeño lo que permite distinguir bien la señal resultante.
- **SNR:** Este indicador mide que tan limpia esta la señal frente al ruido presente en la medición, en este caso arroja un valor de 38.67 dB, lo que indica que la señal está en condiciones de libre ruido y es estable en la medición.
- **ENOB:** Significa número efectivo de bits lo cual a pesar de que el ADC de la Raspberry Pi Pico 2W es de 12 bits a nivel estándar, en la practica no siempre se utilizan los 12 exactos debido a efectos como el ruido o imperfecciones del sistema, por lo cual el valor que nos arroja este indicador es 6.13 bits lo que traduce que solo el ADC empleo el uso de 6 bits útiles.

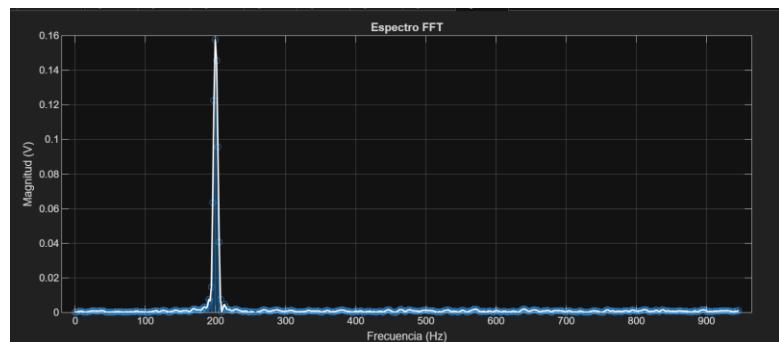
### C. Gráficas de las mediciones obtenidas

En esta sección se emplea MATLAB para graficar los dos archivos generados por el programa. Para el primer archivo, se implementó un código que, a partir de 15 muestras, permitió reconstruir la señal en el dominio del tiempo, obteniendo un resultado satisfactorio. Por otro lado, con el segundo archivo se desarrolló otro código para representar el espectro en frecuencia, utilizando 1024 puntos en la Transformada Rápida de Fourier (FFT), lo que permitió observar con mayor detalle la distribución de la señal en el dominio de la frecuencia.



**Figura 4. Señal senoidal reconstruida en el dominio del tiempo con muestra de 1024**

Como se aprecia en la gráfica anterior, la reconstrucción de la señal fue óptima, lo que confirma que el microcontrolador logró capturarla correctamente. Además, se observa que la señal no está centrada en cero, sino que oscila aproximadamente entre 0.8 V y 2.2 V, presentando un nivel DC cercano a 1.5–1.6 V.



**Figura 5. Espectro en frecuencia reconstruido con puntos de 1024**

En esta gráfica se observa un único armónico claramente definido en 200 Hz, con una magnitud aproximada de 0.158 V. Este pico corresponde a la componente fundamental de la señal senoidal, lo que confirma que se trata de una señal pura con un solo armónico presente en todo el espectro de frecuencia. Asimismo, el piso de ruido se mantiene cercano a cero, con apenas pequeñas variaciones.

#### D. Señal senoidal en distintas frecuencias y FFT

En esta sección se trabajó con tres frecuencias: 300 Hz, 900 Hz y 1800 Hz, configuradas en el generador de funciones. Al mismo tiempo, el número de puntos de la FFT se ajustó en el código suministrado por el docente. Como resultado, se generaron los archivos correspondientes en el dominio del tiempo y de la frecuencia, los cuales posteriormente fueron graficados en MATLAB.

##### - Frecuencia 300 Hz y N\_FFT = 64

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.29 Hz
Offset DC removido: 1.621 V
Frecuencia dominante: 295.98 Hz
Amplitud señal: 0.031 V
Piso de ruido: 0.00352 V (-59.44 dB FS)
SNR: 18.95 dB, ENOB: 2.86 bits
```

Figura 6. Información mostrada por el programa

##### - Frecuencia 300 Hz y N\_FFT = 128

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.70 Hz
Offset DC removido: 1.621 V
Frecuencia dominante: 296.05 Hz
Amplitud señal: 0.108 V
Piso de ruido: 0.00705 V (-53.41 dB FS)
SNR: 23.69 dB, ENOB: 3.64 bits
```

Figura 7. Información mostrada por el programa

##### - Frecuencia 300 Hz y N\_FFT = 256

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.59 Hz
Offset DC removido: 1.617 V
Frecuencia dominante: 303.43 Hz
Amplitud señal: 0.274 V
Piso de ruido: 0.00894 V (-51.34 dB FS)
SNR: 29.72 dB, ENOB: 4.64 bits
```

Figura 8. Información mostrada por el programa

##### - Frecuencia 300 Hz y N\_FFT = 512

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.58 Hz
Offset DC removido: 1.620 V
Frecuencia dominante: 299.73 Hz
Amplitud señal: 0.310 V
Piso de ruido: 0.00261 V (-62.03 dB FS)
SNR: 41.50 dB, ENOB: 6.60 bits
```

Figura 9. Información mostrada por el programa

##### - Frecuencia 300 Hz y N\_FFT = 1024

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.56 Hz
Offset DC removido: 1.618 V
Frecuencia dominante: 299.72 Hz
Amplitud señal: 0.155 V
Piso de ruido: 0.00156 V (-66.52 dB FS)
SNR: 39.96 dB, ENOB: 6.35 bits
```

Figura 10. Información mostrada por el programa

##### - Frecuencia 300 Hz y N\_FFT = 2048

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.65 Hz
Offset DC removido: 1.621 V
Frecuencia dominante: 300.67 Hz
Amplitud señal: 0.080 V
Piso de ruido: 0.00086 V (-71.65 dB FS)
SNR: 39.30 dB, ENOB: 6.24 bits
```

Figura 11. Información mostrada por el programa

#### - Visualización de las señales en el dominio tiempo

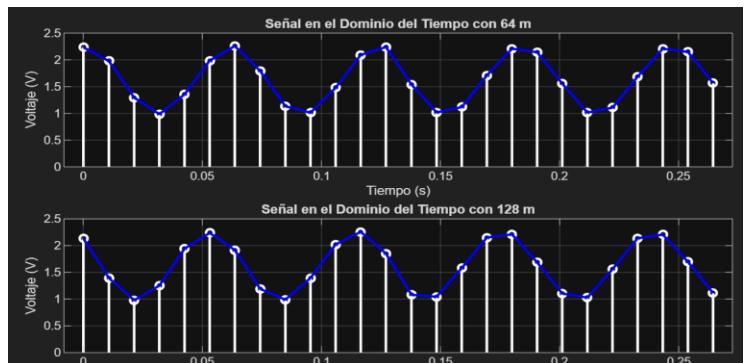
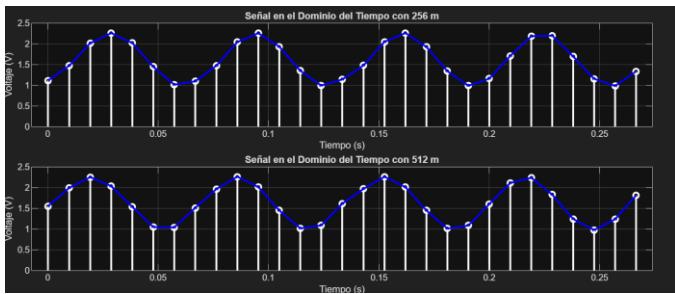


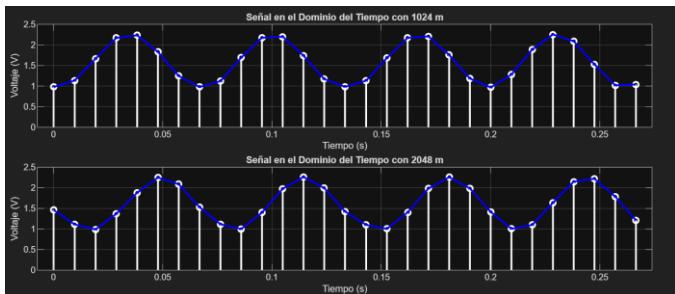
Figura 12. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 64 y 128

Al observar las gráficas, se nota que la señal oscila dentro de un rango aproximado entre 1 V y 2.3 V, con un nivel DC cercano a 1.5 V. Al comparar los dos casos, se aprecia que la gráfica obtenida con 64 muestras presenta una oscilación más irregular, mientras que la de 128 muestras muestra un trazo mucho más suave y definido en cada ciclo. Esto hace que la representación de la señal sea más clara y precisa.

Con base en lo mencionado anteriormente, al aumentar el número de muestras en las señales del dominio del tiempo se puede deducir que la señal se representa de manera más estable. Esto permite observar variaciones en su amplitud de voltaje a medida que incrementa la cantidad de muestras. A continuación, se presentan las gráficas correspondientes para 256, 512, 1024 y 2048 muestras.

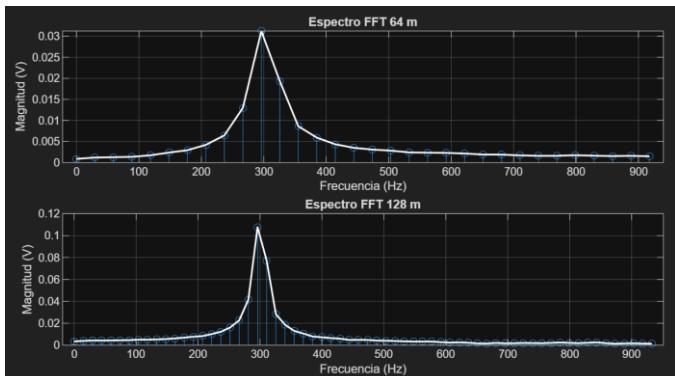


**Figura 13. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 256 y 512**



**Figura 14. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 1024 y 2048**

#### - Visualización del espectro en frecuencia



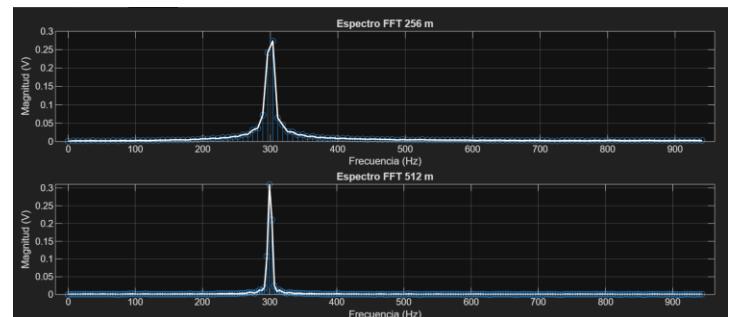
**Figura 15. Espectros en frecuencia reconstruidos con puntos de 64 y 128**

En las gráficas del espectro FFT con 64 y 128 muestras se nota una diferencia importante en la forma del pico principal. En la primera gráfica (64 muestras), el pico aparece más ancho, lo que refleja la baja cantidad de puntos utilizados y hace que la señal se vea menos definida. En cambio, en la segunda gráfica (128 muestras), el pico es mucho más angosto y definido, lo que la convierte en una representación más confiable. Además, en la primera gráfica se observa un pequeño piso de ruido, mientras que en la segunda este piso está mucho más cercano a cero, lo que hace que la señal se vea más clara. Si miramos la magnitud, la diferencia también es notable: en la segunda gráfica alcanza aproximadamente 0.10 V, mientras que en la primera apenas llega a 0.03 V. Esto

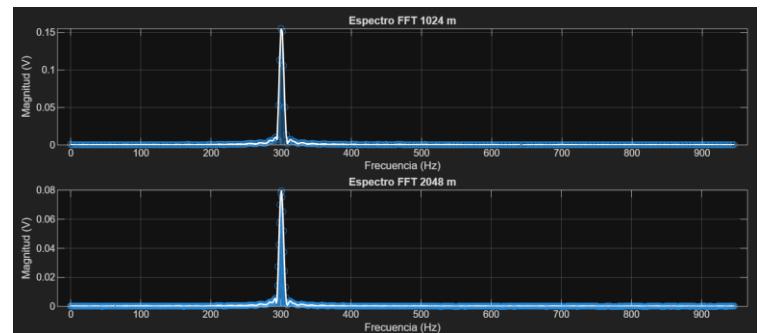
evidencia que con más muestras la señal no solo se representa con mayor precisión, sino que también se conserva mejor la energía.

Ambas gráficas están centradas en la misma frecuencia fundamental, alrededor de 300 Hz, pero la de 128 muestras es claramente más exacta y detallada.

Por último, para obtener un espectro en frecuencia más definido, se incrementa el número de puntos de la FFT. De esta manera, el pico principal se representa con mayor precisión y, al mismo tiempo, el valor de la magnitud presenta ligeras variaciones. Asimismo, el piso de ruido se aproxima más a cero, lo que hace que la señal en el dominio de la frecuencia sea más estable y limpia. A continuación, se muestran los espectros correspondientes para 256, 512, 1024 y 2048 puntos.



**Figura 16. Espectros en frecuencia reconstruidos con puntos de 256 y 512**



**Figura 17. Espectros en frecuencia reconstruidos con puntos de 1024 y 2048**

- Frecuencia 900 Hz y N\_FFT = 64

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.56 Hz
Offset DC removido: 1.434 V
Frecuencia dominante: 917.68 Hz
Amplitud señal: 0.030 V
Piso de ruido: 0.00169 V (-65.84 dB FS)
SNR: 25.01 dB, ENOB: 3.86 bits
```

**Figura 18. Información mostrada por el programa**

- Frecuencia 900 Hz y N\_FFT = 128

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.72 Hz
Offset DC removido: 1.411 V
Frecuencia dominante: 902.95 Hz
Amplitud señal: 0.112 V
Piso de ruido: 0.00735 V (-53.05 dB FS)
SNR: 23.64 dB, ENOB: 3.63 bits
```

**Figura 19. Información mostrada por el programa**

- Frecuencia 900 Hz y N\_FFT = 256

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.84 Hz
Offset DC removido: 1.394 V
Frecuencia dominante: 903.01 Hz
Amplitud señal: 0.290 V
Piso de ruido: 0.00745 V (-52.93 dB FS)
SNR: 31.80 dB, ENOB: 4.99 bits
```

**Figura 20. Información mostrada por el programa**

- Frecuencia 900 Hz y N\_FFT = 512

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.57 Hz
Offset DC removido: 1.534 V
Frecuencia dominante: 902.88 Hz
Amplitud señal: 0.275 V
Piso de ruido: 0.00465 V (-57.02 dB FS)
SNR: 35.43 dB, ENOB: 5.59 bits
```

**Figura 21. Información mostrada por el programa**

- Frecuencia 900 Hz y N\_FFT = 1024

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.93 Hz
Offset DC removido: 1.582 V
Frecuencia dominante: 901.20 Hz
Amplitud señal: 0.151 V
Piso de ruido: 0.00243 V (-62.65 dB FS)
SNR: 35.86 dB, ENOB: 5.66 bits
```

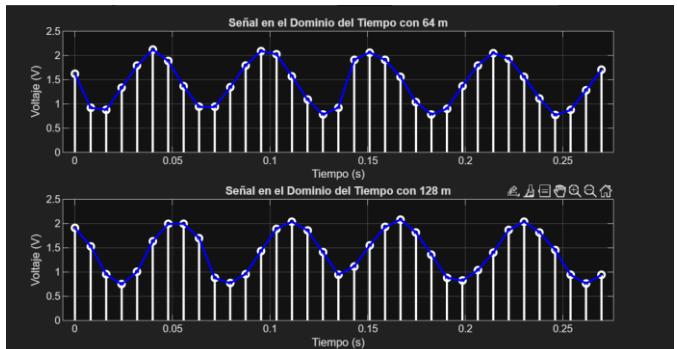
**Figura 22. Información mostrada por el programa**

- Frecuencia 900 Hz y N\_FFT = 2048

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.84 Hz
Offset DC removido: 1.555 V
Frecuencia dominante: 901.16 Hz
Amplitud señal: 0.076 V
Piso de ruido: 0.00135 V (-67.75 dB FS)
SNR: 34.96 dB, ENOB: 5.51 bits
```

**Figura 23. Información mostrada por el programa**

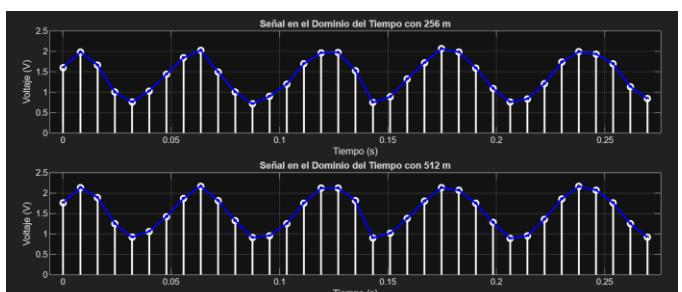
- Visualización de las señales en el dominio tiempo



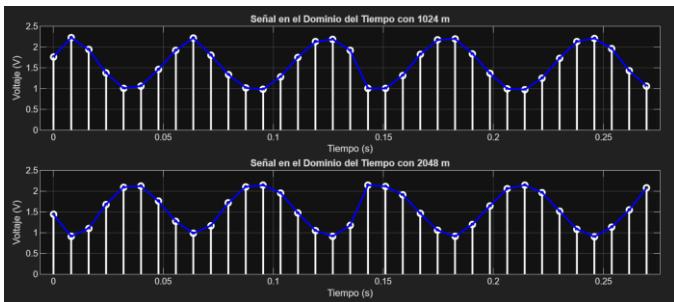
**Figura 24. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 64 y 128**

Al observar las gráficas, se evidencia que la señal con 128 muestras presenta una forma más suave y definida en el dominio del tiempo, lo que la hace más confiable. En contraste, la señal con 64 muestras muestra cierta distorsión, con oscilaciones que en algunos puntos se ven irregulares o “torcidas”, lo que reduce la precisión de su representación. Por lo tanto, al aumentar el número de muestras como en el caso de 128 la señal se interpreta con mayor claridad.

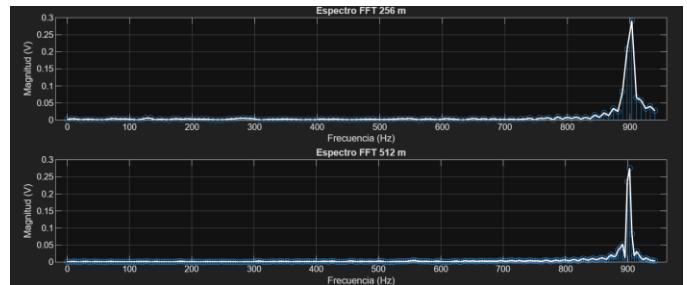
Con base en lo mencionado anteriormente, se puede afirmar que, a medida que aumenta el número de muestras, las señales en el dominio del tiempo se representan de manera más definida. A continuación, se presentan los resultados para 256, 512, 1024 y 2048 muestras.



**Figura 25. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 256 y 512**

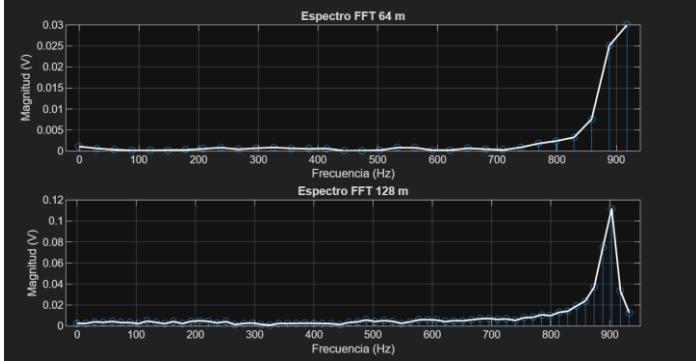


**Figura 26. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 1024 y 2048**



**Figura 28. Espectros en frecuencia reconstruidos con puntos de 256 y 512**

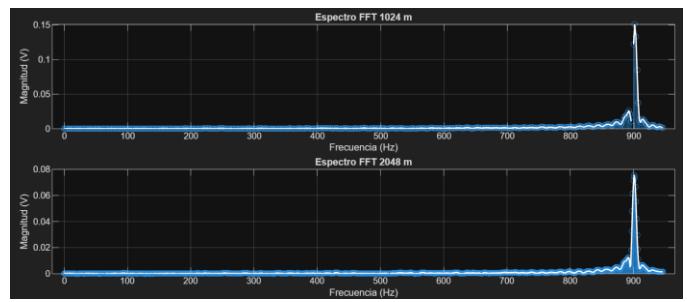
#### - Visualización del espectro en frecuencia



**Figura 27. Espectros en frecuencia reconstruidos con puntos de 64 y 128**

Al comparar las dos gráficas, se observa que en la primera aparece un pico de energía alrededor de los 900 Hz, pero su magnitud apenas alcanza los 0.03 V debido a la baja cantidad de puntos utilizados en la FFT. Esto hace que el espectro se vea con poca definición, dificultando la identificación precisa de las frecuencias presentes. En cambio, en la segunda gráfica el pico es mucho más claro y definido, con una magnitud cercana a 0.11 V. Gracias al doble de muestras, la resolución del espectro mejora notablemente, mostrando una representación más suave, precisa.

Por consiguiente, al incrementar el número de puntos en la FFT, el espectro en el dominio de la frecuencia alrededor de los 900 Hz se observa con mayor definición, ya que el pico correspondiente se distingue de forma más precisa. En cuanto a la magnitud, se presentan leves variaciones en el nivel de amplitud y a nivel de piso de ruido mucho más atenuado a cero. A continuación, se muestran los espectros obtenidos para 256, 512, 1024 y 2048 muestras.



**Figura 29. Espectros en frecuencia reconstruidos con puntos de 1024 y 2048**

#### - Frecuencia de 100 Hz y N\_FFT = 64

```

MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.59 Hz
Offset DC removido: 1.615 V
Frecuencia dominante: 88.81 Hz
Amplitud señal: 0.033 V
Piso de ruido: 0.00361 V (-59.21 dB FS)
SNR: 19.10 dB, ENOB: 2.88 bits

```

**Figura 30. Información mostrada por el programa**

#### - Frecuencia de 100 Hz y N\_FFT = 128

```

MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.63 Hz
Offset DC removido: 1.629 V
Frecuencia dominante: 88.81 Hz
Amplitud señal: 0.107 V
Piso de ruido: 0.00575 V (-55.17 dB FS)
SNR: 25.42 dB, ENOB: 3.93 bits

```

**Figura 31. Información mostrada por el programa**

#### - Frecuencia de 100 Hz y N\_FFT = 256

```

MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.61 Hz
Offset DC removido: 1.630 V
Frecuencia dominante: 96.21 Hz
Amplitud señal: 0.257 V
Piso de ruido: 0.00750 V (-52.87 dB FS)
SNR: 30.71 dB, ENOB: 4.81 bits

```

**Figura 32. Información mostrada por el programa**

- Frecuencia de 100 Hz y N\_FFT = 512

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.48 Hz
Offset DC removido: 1.610 V
Frecuencia dominante: 92.50 Hz
Amplitud señal: 0.258 V
Piso de ruido: 0.00663 V (-53.95 dB FS)
SNR: 31.82 dB, ENOB: 4.99 bits
```

Figura 33. Información mostrada por el programa

- Frecuencia de 100 Hz y N\_FFT = 1024

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.58 Hz
Offset DC removido: 1.608 V
Frecuencia dominante: 92.51 Hz
Amplitud señal: 0.133 V
Piso de ruido: 0.00324 V (-60.15 dB FS)
SNR: 32.29 dB, ENOB: 5.07 bits
```

Figura 34. Información mostrada por el programa

- Frecuencia de 100 Hz y N\_FFT = 2048

```
MPY: soft reboot
Iniciando adquisición y análisis...
Frecuencia deseada: 2000 Hz, frecuencia real: 1894.12 Hz
Offset DC removido: 1.631 V
Frecuencia dominante: 91.56 Hz
Amplitud señal: 0.068 V
Piso de ruido: 0.00170 V (-65.78 dB FS)
SNR: 32.07 dB, ENOB: 5.04 bits
```

Figura 35. Información mostrada por el programa

- Visualización de las señales en el dominio tiempo

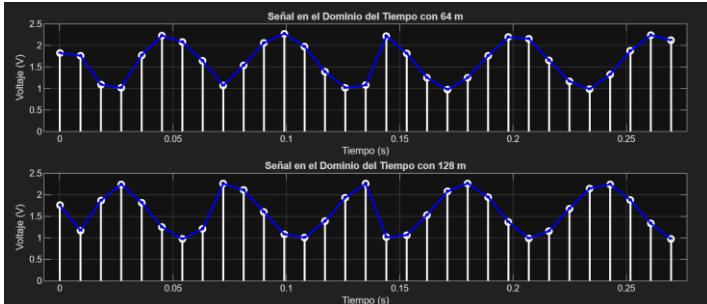


Figura 36. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 64 y 128

Como se observa en las dos gráficas, ambas presentan cierto grado de distorsión en sus oscilaciones, variando aproximadamente entre 1 V y 2.4 V y con un nivel DC cercano a 1.5 V. A estas señales se les aplicó una reducción de 17 muestras, valor que permitió ajustar la forma para aproximarse a una señal senoidal. En cuanto al muestreo, se nota que la segunda gráfica, aunque aún presenta algo de

distorsión, muestra una oscilación más estable y definida en comparación con la primera figura.

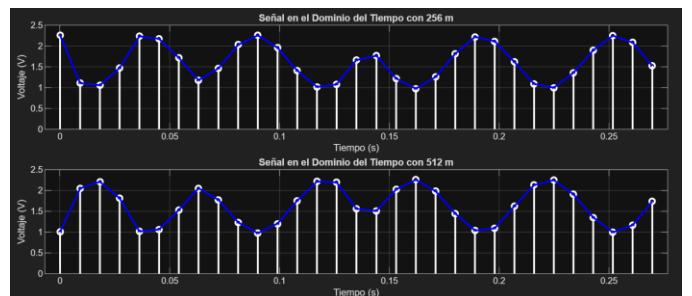


Figura 37. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 256 y 512

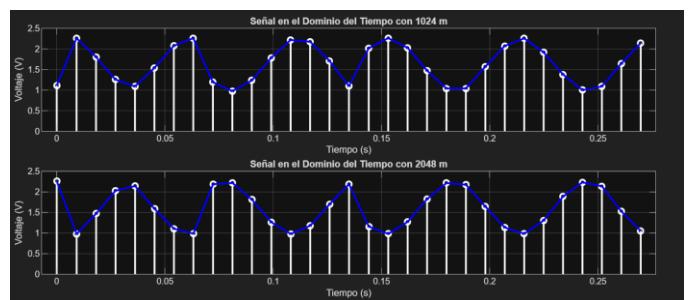


Figura 38. Señales senoidales reconstruidas en el dominio del tiempo con muestras de 1024 y 2048

- Visualización del espectro en frecuencia

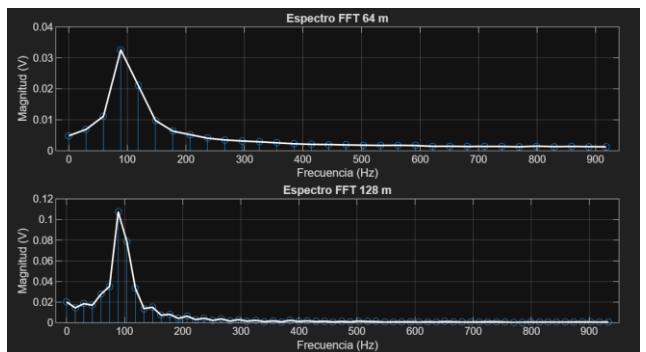
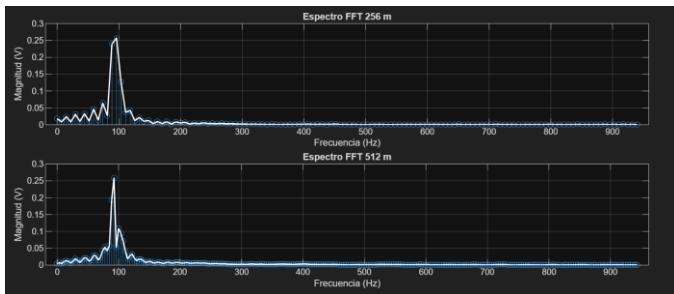
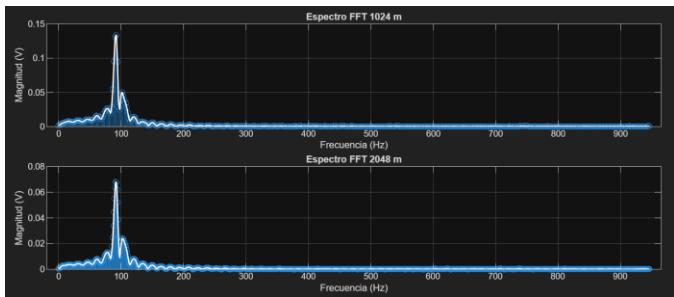


Figura 39. Espectros en frecuencia reconstruidos con puntos de 64 y 128

Al observar las dos gráficas del espectro en frecuencia, se aprecia que la correspondiente a 128 muestras resulta más estable, ya que su pico principal aparece bien definido y alcanza una magnitud aproximada entre 0.10 y 0.11 V. En contraste, en la gráfica de 64 muestras el pico es más ancho y menos preciso debido a la menor cantidad de puntos, registrando una magnitud de apenas 0.031 V. Además, el piso de ruido en la primera gráfica se encuentra más elevado, mientras que en la de 128 muestras está mucho más cercano a cero, lo que permite una representación más clara y confiable de la señal.



**Figura 40. Espectros en frecuencia reconstruidos con puntos de 256 y 512**



**Figura 41. Espectros en frecuencia reconstruidos con puntos de 1024 y 2048**

### E. Ejecución de nuevo código

En este inciso se requirió diseñar un programa diferente al proporcionado por el docente, con el objetivo de muestrear una señal senoidal de 200 Hz, con una amplitud de 1.2 Vpp y una componente DC de 1.6 V. Al código desarrollado se le incorporó la simulación de jitter, entendido como la variación aleatoria en los instantes de muestreo de la señal dentro de un intervalo de tiempo. Además, el programa fue configurado para generar dos archivos: uno con los datos del muestreo de la señal y otro con los valores de jitter registrados en cada intervalo de tiempo. Estos archivos se graficaron posteriormente en MATLAB para realizar el análisis correspondiente.

Es importante señalar que en este código no fue posible generar el espectro mediante FFT, ya que dicha operación resultaba demasiado pesada para el microcontrolador. Esto se debe principalmente a la limitada memoria RAM disponible, lo que restringe la ejecución de procesos de este tipo.

#### - Visualización de la señal en el dominio tiempo



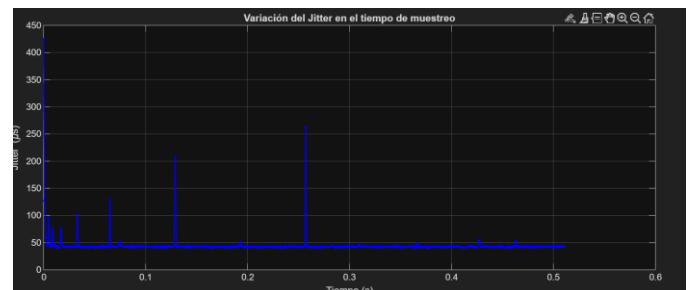
**Figura 42. Señal senoidal reconstruida en el dominio del tiempo**

Al ejecutar el código diseñado, se observa que la señal se genera de manera estable, mostrando una transición en el tiempo entre 0 y 0.27 segundos. En cuanto a la amplitud, esta oscila aproximadamente entre 0.9 V y 2.3 V, con un valor DC cercano a 1.6 V, lo que explica que la señal aparezca desplazada hacia arriba respecto al eje de referencia.

Para este caso, se aplicó una reducción a 20 muestras, lo cual permite visualizar una representación más clara y definida de la señal. Sin embargo, si se disminuye aún más el número de muestras, la forma senoidal original comienza a perderse, adoptando un aspecto más triangular. Esto implica que ya no se cumpliría con la condición inicial de representar adecuadamente una señal senoidal.

Por otro lado se muestra la grafica del jitter, en la cual se ve muestra a continuación.

#### - Visualización del Jitter en el dominio del tiempo



**Figura 43. Gráfica del Jitter en el dominio del tiempo**

Al observar la gráfica, se nota que la mayoría de los valores de jitter se mantienen entre 48 y 100  $\mu$ s, lo que puede considerarse un rango normal para el sistema. Sin embargo, también aparecen algunos picos aislados, cercanos a los 100, 200 e incluso 400  $\mu$ s. Estos picos indican que en ciertos momentos el microcontrolador pudo haber sufrido interrupciones o pequeños retrasos, lo que provocó una mayor variación en los tiempos de muestreo.

## III. ANALISIS DE RESULTADOS

¿Qué es el Jitter? ¿Qué implicaciones tiene en el proceso de codificación de la fuente?

El Jitter es la variación en la periodicidad de una señal o de un evento periódico con respecto a su frecuencia ideal o real. Dicho de otro modo, representa pequeñas desviaciones en el momento exacto en que debería ocurrir un evento.

Por ejemplo, si se desea muestrear una señal cada 100  $\mu$ s, el Jitter puede hacer que algunas muestras se tomen a los 95  $\mu$ s y otras a los 105  $\mu$ s. Esto significa que el Jitter introduce variaciones que pueden adelantarse o retrasarse respecto al instante de muestreo esperado.

## Implicaciones en la codificación de la fuente

### Distorsión de la señal muestreada

- Si las muestras no se toman en el momento preciso, la reconstrucción de la señal original se ve afectada.
- Esto genera errores en la forma de onda, especialmente en señales de alta frecuencia

### Pérdida de calidad en la información

- En audio digital, el Jitter produce ruidos o distorsiones.
- En video digital, puede generar pérdida de nitidez o artefactos.

### Limitación del desempeño del sistema

- En microcontroladores o sistemas embebidos, el aumento del Jitter significa menos precisión en el muestreo, lo que implica aplicaciones sensibles como sensores de tiempo real.

¿Qué alternativas existen con el dispositivo Raspberry Pi pico 2W para realizar un muestreo exitoso considerando las características técnicas y posibilidades del dispositivo?

La Raspberry Pi Pico 2W brinda diferentes alternativas para optimizar el muestreo de señales. Una de ellas es aprovechar el subsistema PIO (Programmable I/O), que permite generar pulsos de muestreo con gran precisión y menor nivel de jitter. También puede recurrirse al uso de DMA (Direct Memory Access), que posibilita transferir directamente los datos del ADC hacia la memoria sin necesidad de que la CPU intervenga en cada operación, lo que disminuye la latencia y libera capacidad de procesamiento. Además, ajustar de forma adecuada los parámetros del ADC y utilizar buffers circulares contribuye a mantener un flujo de datos más estable y continuo, ofreciendo un desempeño más cercano al de un sistema dedicado de adquisición de señales.

## IV. CONCLUSIONES

- El muestreo de señales no solo depende de la frecuencia a la que se toman los datos, sino también de la estabilidad del tiempo de muestreo. El jitter, aunque pueda parecer pequeño, puede cambiar la forma en que una señal es reconstruida.
- El análisis de las gráficas de tiempo y frecuencia permitió evidenciar cómo el aumento del número de muestras mejora la definición de la señal, tanto en el dominio temporal como en el espectro, reduciendo la forma de onda.

- El estudio del Jitter mostró que la mayoría de los valores se concentran en un rango bajo y estable, aunque se presentaron picos aislados que revelan interrupciones o retardos en el microcontrolador. Estos eventos esporádicos son clave para entender las limitaciones del sistema.

- Se observó que, al disminuir el número de muestras, la señal empieza a perder su forma senoidal y tiende a aproximarse a una señal triangular, lo que confirma la importancia de elegir un muestreo adecuado para preservar las características originales de la señal.

## V. REFERENCIAS

- [1]. Texas Instruments. (s.f.). *¿Qué es el jitter y por qué importa?* Cirrus Logic. Recuperado de <https://www.cirrus.com/company/innovation/what-is-jitter>
- [2]. Raspberry Pi. (s.f.). *Documentación oficial del SDK del Raspberry Pi Pico: APIs de hardware.* Recuperado de <https://www.raspberrypi.com/documentation/pico-sdk/hardware.html>
- [3]. Twilio. (s.f.). *Qué es el jitter.* Recuperado de <https://www.twilio.com/docs/glossary/what-is-jitter>
- [4]. Analog Devices. (s. f.). Maximum SNR vs. Clock Jitter. Analog Devices. Recuperado el 9 de septiembre de 2025, de <https://www.analog.com/en/resources/technical-articles/maximum-snr-vs-clock-jitter.html>
- [5]. Iosoft. (2021, 26 de octubre). *Entrada ADC en Raspberry Pi Pico usando DMA y MicroPython.* Recuperado de <https://iosoft.blog/2021/10/26/pico-adc-dma/>