1. Como el algoritmo esta implementado para potencias de 2 tarda menos por lo cual es más eficiente, y cuando no tengo una potencia de 2 se rellena con 0, por lo cual tiene más costo y es más lento.

Con la prueba obtenida, se puede ver al final que la muestra 2 tiene muchas más frecuencias que la muestra 1.

* + - **Muestra 1:** Se pueden observar los picos altos de igual magnitud en los puntos -0.4 y 0.4.
    - **Muestra 2:**  Se pueden observar los picos altos de la gráfica de igual magnitud en 2 puntos muy cercanos a 0.

1. Lo que se podría decir es que la muestra 2 se podría esperar frecuencias muchas mas altas por tener más datos.
2. Se podría decir que la Muestra 2 es mejor solución debido a que es más rápida que la Muestra 1, por lo cual más rápido completara el ciclo.
3. 1. En Numpy, es **fft2** la función para calcular la transformada de Fourier de una imagen que recibe un arreglo bidimensional (imagen) y devuelve las frecuencias en img\_c2

En img\_c3 se utiliza la función **fftshift** que se encarga de centrar las frecuencias más altas.

En img\_c4 se utiliza **fffishift** la cual permite revertir el ultimo cambio realizado en el paso anterior

Por último se utiliza **ifft2** para calcular la inversa de fft2, reconstruyendo la imagen después de procesarla.

* 1. El filtro de paso bajo es un filtro que solo permite el paso de frecuencias bajas. Las bajas frecuencias en las imágenes significan valores de píxeles que cambian lentamente.

Dado esto, los contenidos de las frecuencias altas, como los ruidos, se bloquean, lo que hace que la imagen procesada tenga píxeles menos ruidosos. Por lo tanto, el filtro de paso bajo se utiliza mucho para eliminar los ruidos en las imágenes.

* 1. El filtro de paso bajo tiende a preservar la información general de una imagen. Por otro lado, el filtro de paso alto intenta identificar cambios en una imagen.

1. 1. **numpy.fft** promueve arreglos float32y complex64 para arreglos float64y complex128 respectivamente.

<https://numpy.org/doc/stable/reference/routines.fft.html>

* 1. El módulo SciPy **scipy.fftes** un superconjunto más completo numpy.fft, que incluye solo un conjunto básico de rutinas. El paquete scipy.fftpack es una implementación de FFT que no promueve matrices de entrada.

[https://docs.scipy.org/doc/scipy/reference/reference/fftpack.html#module-scipy.fftpack](https://docs.scipy.org/doc/scipy/reference/reference/fftpack.html%23module-scipy.fftpack)