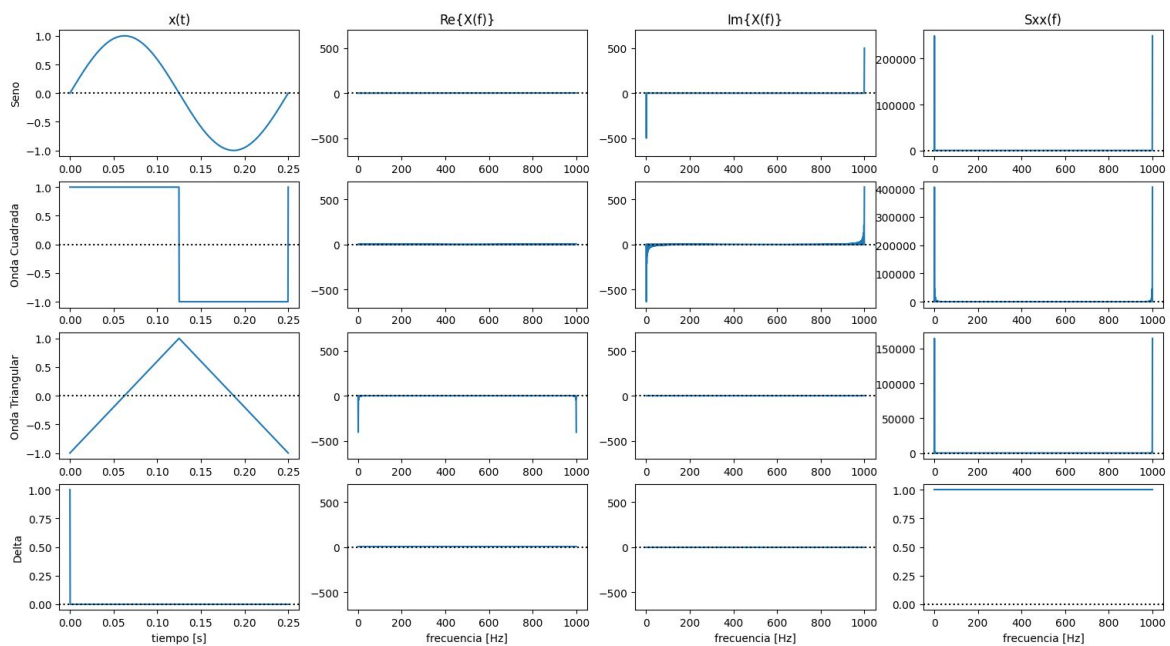


# TRABAJO PRÁCTICO 2

## DFT

1. Grafique las siguientes señales lado a lado con su respectivo espectro en frecuencias:

- senoidal
- cuadrada
- triangular
- delta en  $t=0$



El ancho de banda si bien no se ve en la imagen no es limitado para la onda rectangular y la triangular. Con respecto al ancho de banda de la señal delta es infinito por que un cambio abrupto de abajo hacia arriba y de arriba a abajo en tiempo 0 produce componentes en todas las frecuencias.

Indicando en cada caso los parámetros destacados como:

- frecuencia
- amplitud
- densidad espectral de potencia
- $F_s$
- $N$
- $B$

Señal	Potencia	Valor calculado
Sinusoidal	$A^2/2$	0.5
Cuadrada	$A^2$	1
Triangular	$A^2/3$	0.33
Delta	1	1

El valor de potencia obtenido de la onda cuadrada se entiende que es mayor el analítico con respecto al calculado mediante la  $S_{xx}(f)$  debido a que es una señal que posee potencia en más frecuencias que sólo en sus máximos por eso tenemos esa diferencia.

Con respecto a la señal delta sucedió lo mismo se tomaron dos puntos máximos como si fuera una sinusoidal y por lo tanto el error es mucho mayor. En este caso posee un espectro continuo y su potencia es igual a 1.

Los valores obtenidos experimentalmente muestran el valor de potencias realizando la suma de dos máximos y escalando el resultado con respecto a la frecuencia de muestreo, por lo cual se pueden ver los errores del cálculo de potencia para cada caso:

```

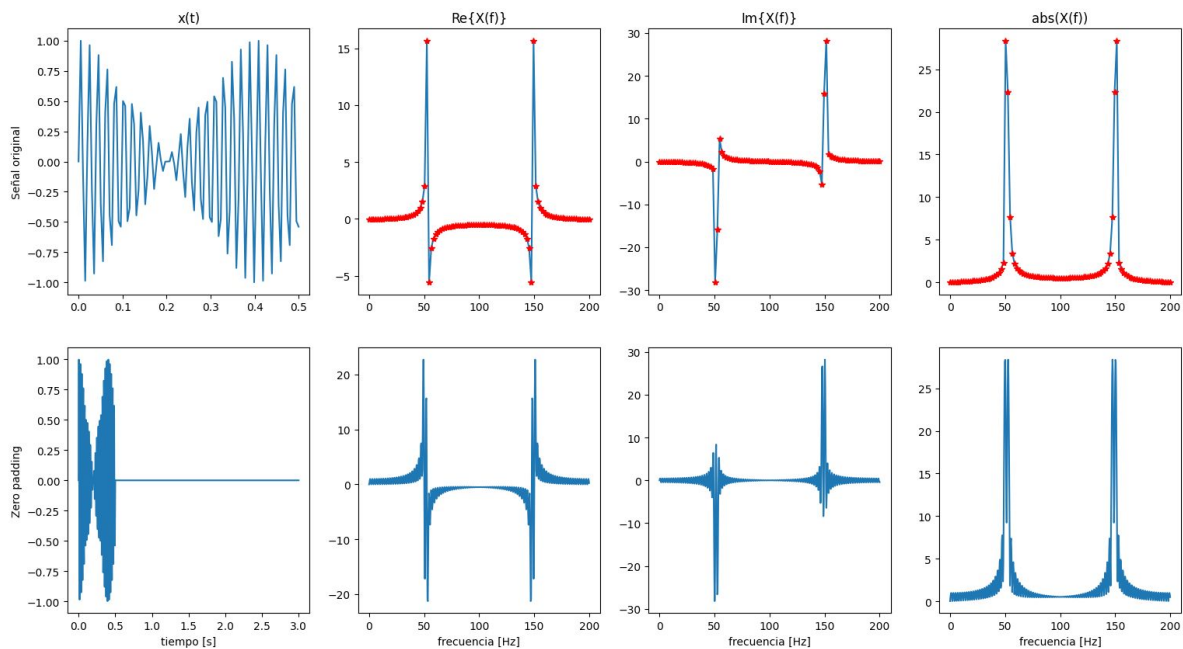
Frecuencia señales: 4
Frecuencia de muestreo: 1000
Cantidad de muestras: 1000
Amplitud: 1
-----Senoidal-----
Max( $S_{xx}(f)$ ) = 249749.11256522444
Potencia: 0.4994982251304489
-----Cuadrada-----
Max( $S_{xx}(f)$ ) = 405282.0679053163
Potencia: 0.8105641358106326
-----Triangular-----
Max( $S_{xx}(f)$ ) = 164584.17946863887
Potencia: 0.32916835893727775
-----Delta-----
Max( $S_{xx}(f)$ ) = 1.0
Potencia: 2e-06

```

El programa en python asociado a este ejercicio se puede ver en este [link](#).

1. Dada la siguiente secuencia de números con  $N=100$  y  $F_s=200$ , indique:

- Resolución espectral
- Obtenga el contenido espectral
- Que técnica conoce para mejorar la resolución en frecuencia?
- Aplique la técnica, grafique y comente los resultados



El agregado de ceros a una señal denominado comúnmente zero padding produce que se posean mayor cantidad de elementos de la señal medida, aunque estos sean ficticios y no posean valor, provocan que la señal transformada posea mayor resolución ya que los saltos en frecuencias están dados por el cociente entre la frecuencia de muestreo y la cantidad de elementos. A los fines prácticos de la resolución del ejercicio se utilizaron además de los 100 elementos del vector provisto, 500 elementos más para el zero padding, debiera ser un múltiplo de  $2^N$  para que la fft sea más eficiente.

Resolución espectral de la señal original:

- $f_s/N = 2\text{Hz}$

Resolución espectral utilizando zero padding ( $N_{zp} = N + 5*N$ ):

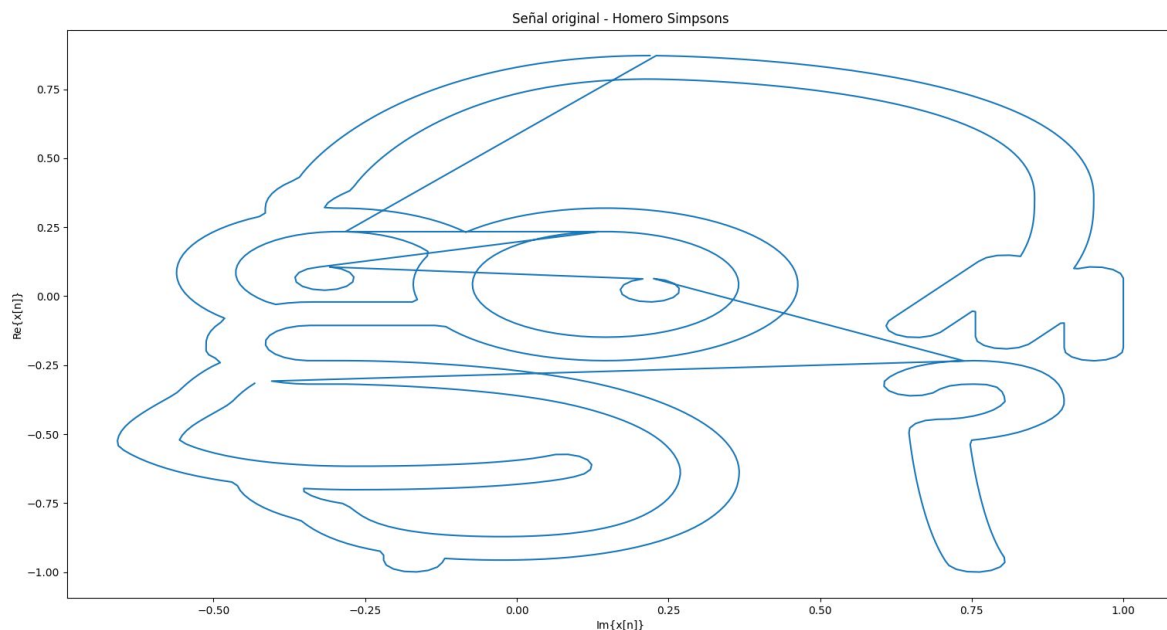
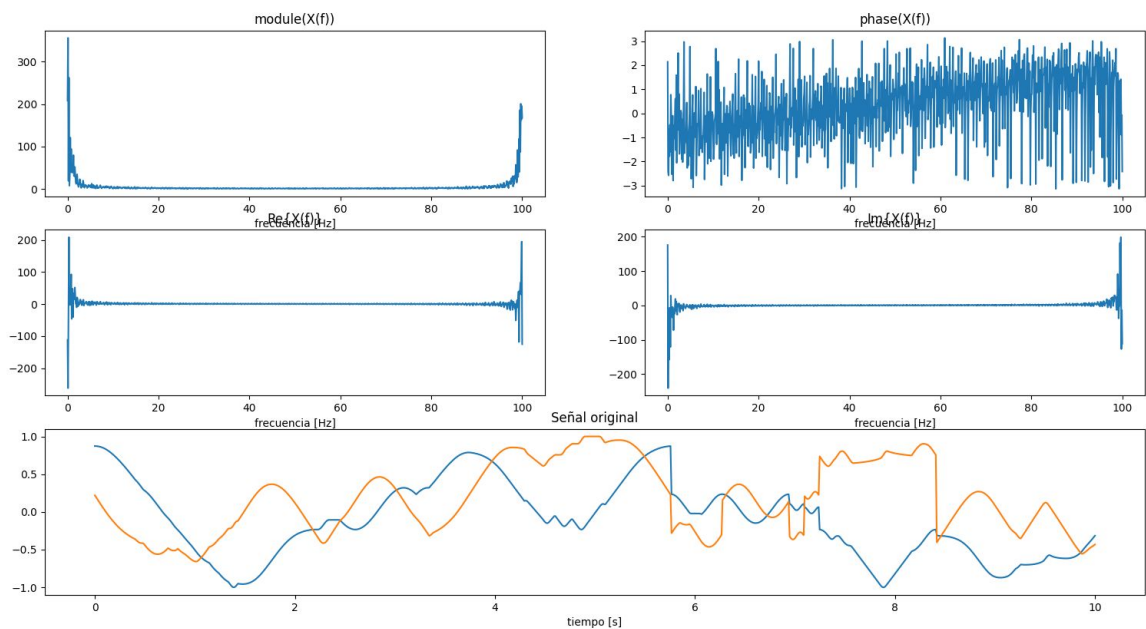
- $f_s(N_{zp}) = 0.33\text{Hz}$

El programa en python asociado a este ejercicio se puede ver en este [link](#).

## iDFT

Dado el siguiente espectro extraído del archivo `fft_hjs.npy`, indique:

- Que cree que representa esta señal? tip: grafique en 2d la idft
- Hasta qué punto podría limitar el ancho de banda y que se siga interpretando su significado
- Grafique para mostrar los resultados



Recortando el espectro en 5 a 6 Hz y sampleando mínimo a 12 muestras por segundo se puede reproducir el mismo gráfico.

El programa en python asociado a este ejercicio se puede ver en este [link](#).

## Convolución

Dado el segmento de audio en el archivo `chapu_noise.npy` con `fs=8000` y sumergido en ruido de alta frecuencia resuelva:

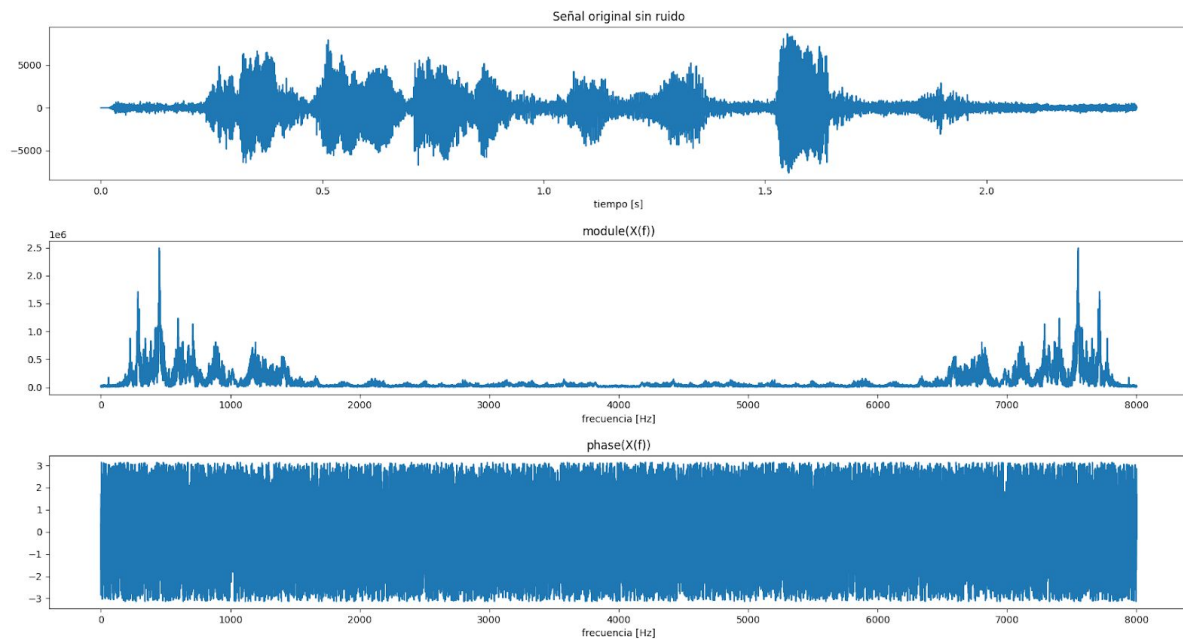
Diseñe un filtro que mitigue el efecto del ruido

Grafique el espectro antes y después del filtro

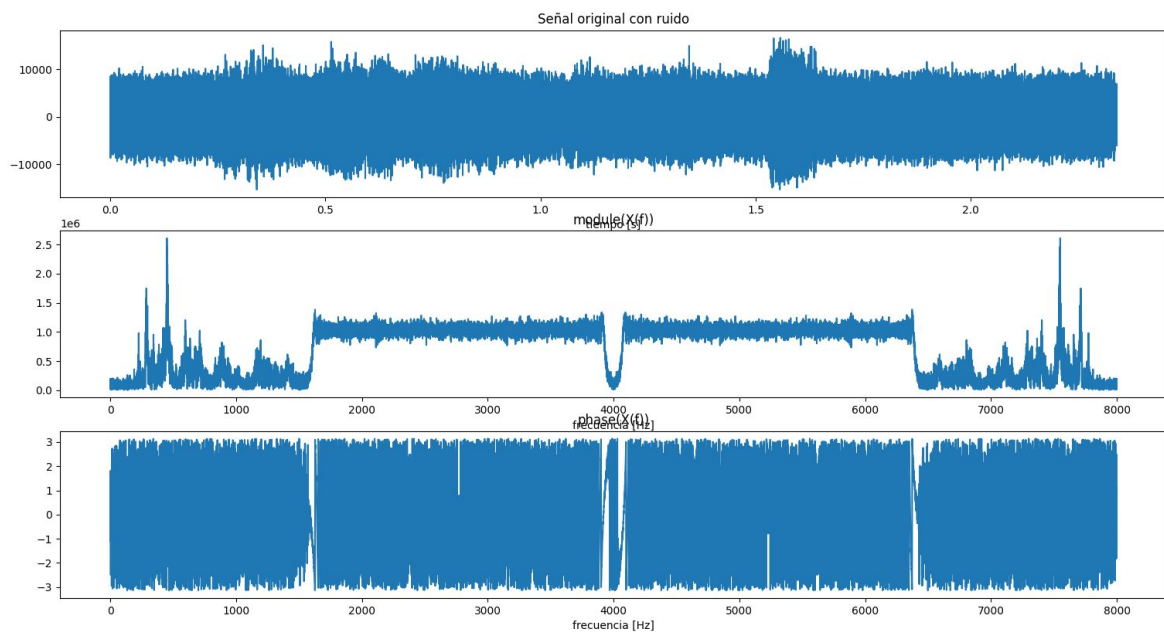
Reproduzca el segmento antes y después del filtrado

Comente los resultados obtenidos

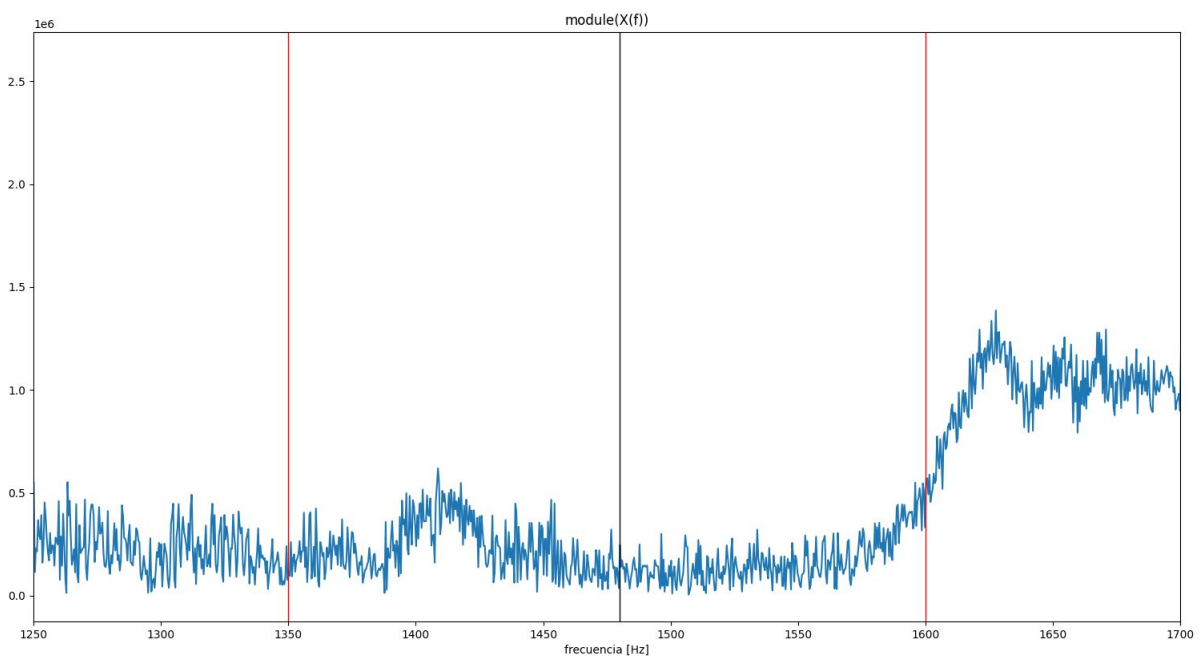
La siguiente imagen muestra la señal de audio original sin ruido incorporado:



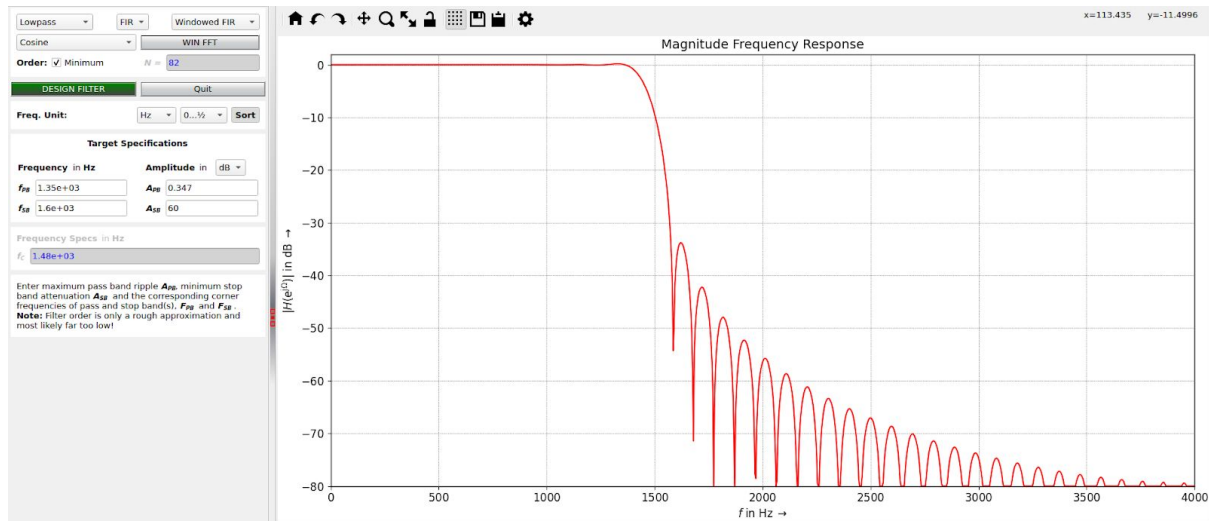
La señal que debemos procesar posee el siguiente gráfico en tiempo y también se puede ver tanto el módulo del espectro como la fase del mismo:



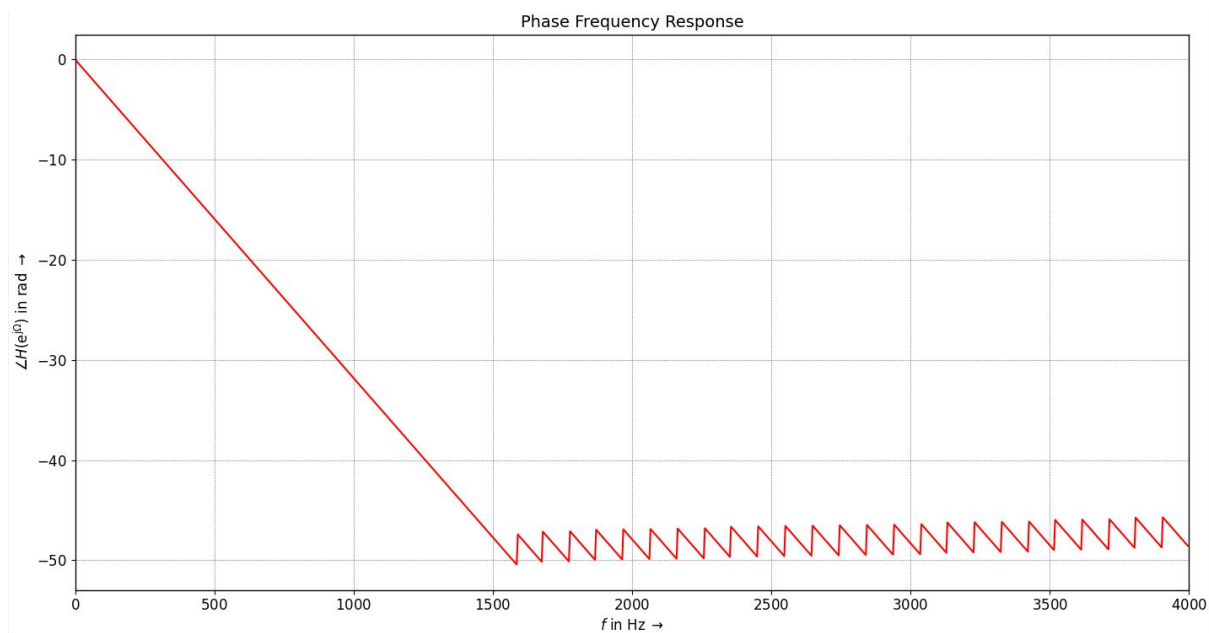
Haciendo zoom en el módulo del espectro en frecuencia se obtiene la forma de onda del mismo con mayor resolución se determinaron los límites de inferior y superior del filtro en frecuencia de 1350 y 1600 Hz respectivamente, cuya frecuencia de corte es de 1480 Hz. Se pretende perder un poco de calidad en el filtrado de la señal aumentando su ancho de banda para poder utilizar un filtro menos abrupto y por lo cual los cambios en fase no sean muy importantes:



Utilizando el programa pyfda se determinó el filtro mínimo que cumple estas características para un valor de rechazo de 60 dB.



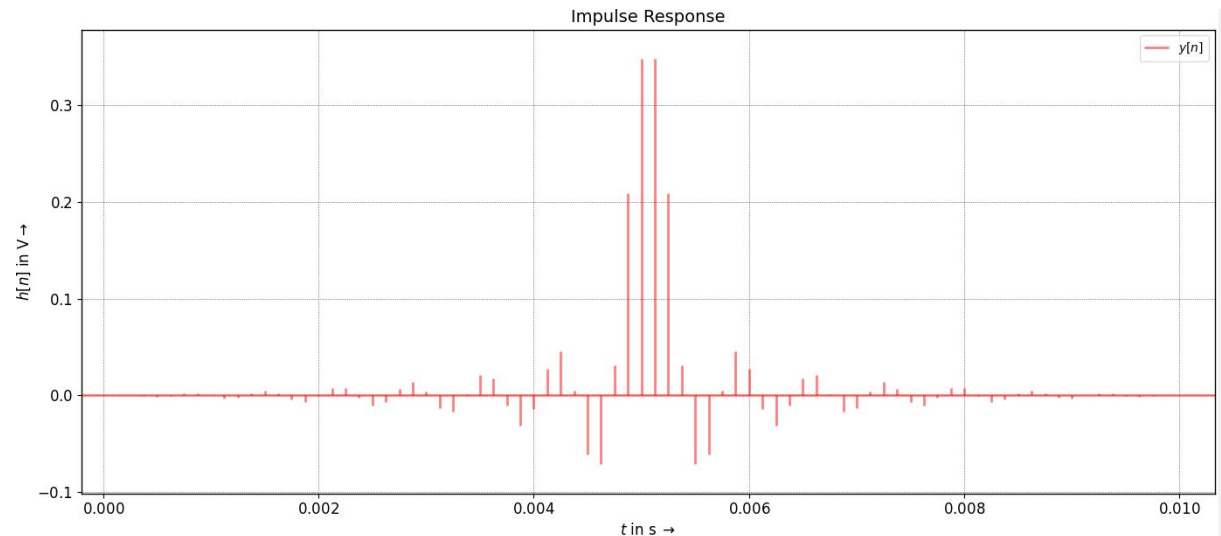
Como puede verse la fase del filtro no varía en demasía, esto se debe principalmente al tipo de filtro empleado FIR (Cosine), al ancho de banda seleccionado y al número de elementos utilizado (mínimo). Si se aumentaba el número de elementos la banda de rechazo era más abrupta el el corrimiento de fase también, por lo cual se determinó que el filtro logrado era el que mejor se adapta a estas situaciones.



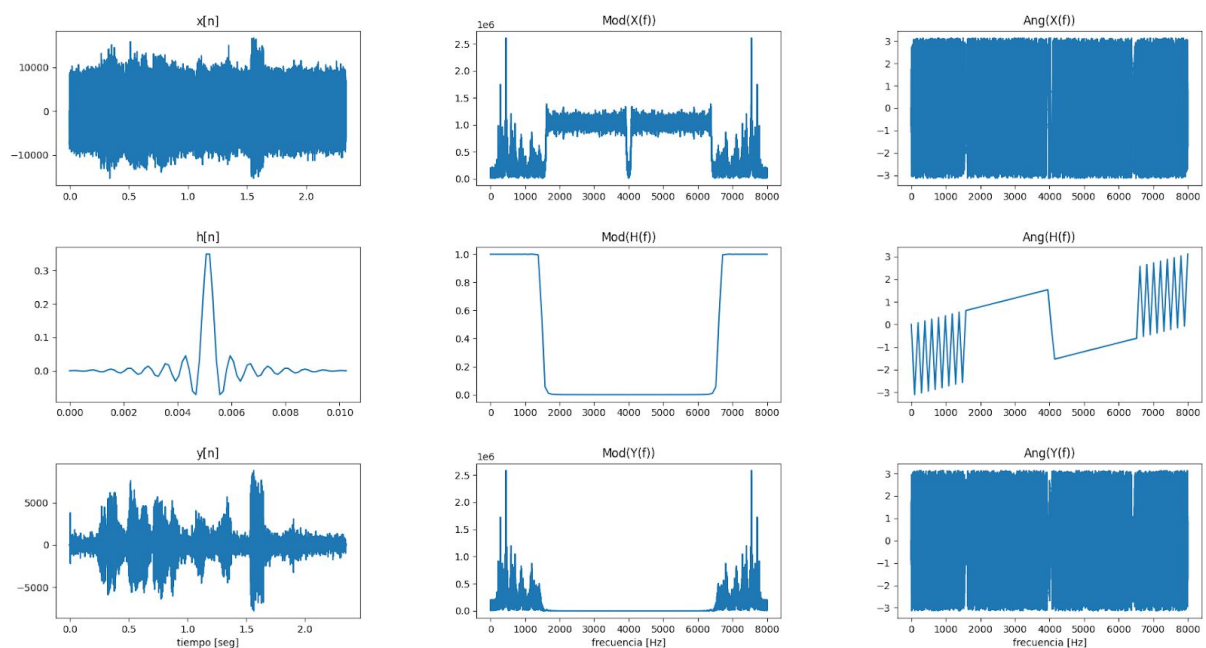
Utilizando los datos obtenidos de este filtro se realiza el filtrado mediante convolución obteniéndose  $N+M-1$  elementos:

```
Frecuencia de muestreo: 8000
Elementos x(n): 18713
Elementos h(n): 82
Elementos y(n): 18794
```

Respuesta al impulso  $h[n]$  en el tiempo:



El siguiente gráfico muestra la señal  $x(n)$ ,  $h(n)$  y el resultado de la convolución  $y(n)$  cada una de ellas junto con su módulo y fase en el espectro en frecuencias:



Si bien se puede ver que la señal original es muy parecida a la obtenida tanto en el tiempo como en frecuencia (módulo y fase) al reproducirla no se obtuvieron buenos resultados, se desconoce la causa de ello.

El programa en python asociado a este ejercicio se puede ver en este [link](#).