**UNIVERSIDAD DE SANTIAGO DE CHILE**
**FACULTAD DE INGENIERÍA**
**DEPARTAMENTO DE INGENIERÍA INFORMÁTICA**

# Laboratorio 1: Señales y Fourier



| | |
|---|---|
| Integrantes: | Francisco Rousseau |
| Curso: | Redes de computadores |
| Profesor: | Carlos González |
| Ayudante: | Nicole Reyes |



21 de Mayo de 2020

# Tabla de contenidos

# 1. Introducción

## 1.1. Objetivos

El objetivo principal de este laboratorio es comprender los fundamentos de la teoría de la comunicación, tales como la existencia y composición de señales análogas y digitales, como también la transformada de Fourier junto con sus usos y propiedades.

## 1.2. Experiencia

Para la realización de lo anteriormente mencionado se realizará un programa desarrollado en el lenguaje de programación Python en conjunto con las librerías scipy, numpy y matplotlib; dicho programa recibirá como entrada el archivo "handel.wav"sobre el cual se realizarán distintas operaciones teniendo como salida gráficos y otro archivo de audio. En esta experiencia se utilizarán

# 2. Marco Teórico

## 2.1. Señales Analógicas

Las señales analógicas son definidas por la Universidad Nacional de Rosario en su publicación **"Electrónica III"**(Rosario) como *variables eléctricas que evolucionan en el tiempo en forma análoga a alguna variable física, que pueden presentarse en forma de corriente, tensión o una carga eléctrico y que varía en forma continua entre un límite inferior y un límite superior.*

Por lo tanto, la señal análoga es aquella que transmite una información que puede ser representada con una función matemática continua, dichas señales son adecuadas para transmitir elementos multimedia (audio y video), expandiéndose mediante ondas senoidales (Ver Figura 1.), pudiendo ser solo leídas por dispositivos que estén diseñados para este fin específico, vale decir, dispositivos análogos.

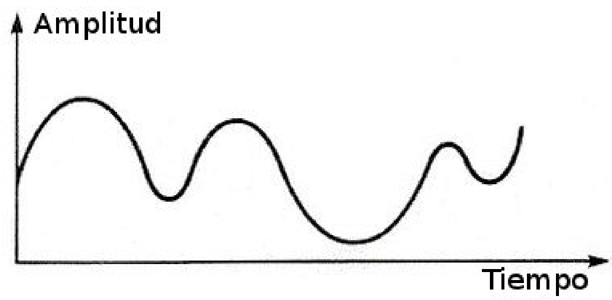


Figura 1: Ejemplo de señal analógica

## 2.2. Señales Digitales

Por otro lado, la UNR define a las señales digitales como *"variables eléctricas con dos niveles bien diferenciados que se alternan en el tiempo transmitiendo información según un código previamente acordado. Cada nivel eléctrico representa uno de dos símbolos: 0 ó 1, V o F esto otorga la capacidad de pasar de un valor a otro sin atravesar los valores intermedios."*

Los sistemas que emplean señales digitales suelen apelar a la lógica binaria, de dos estados, los cuales son reemplazados por unos y ceros, que indican el estado, alto o bajo en el nivel de tensión eléctrica. Estas señales pierden poca calidad y pueden reconstruirse mediante un procedimiento de restauración.

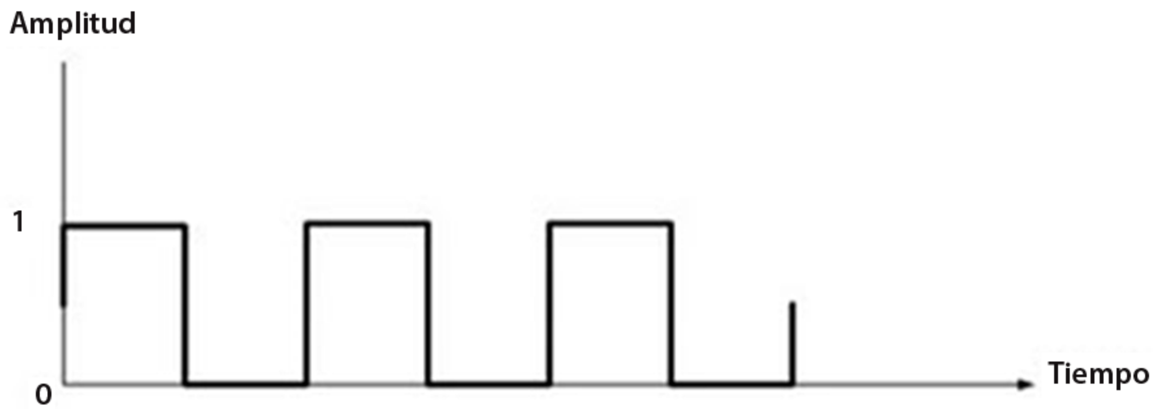


Figura 2: Ejemplo de señal digital

## 2.3. Transformada de Fourier

Finalmente, la UNR define a la transformada de fourier de una señal f(x) continua como una *"transformación que permite calcular el aporte de cada valor de frecuencia a la conformación de la señal."*

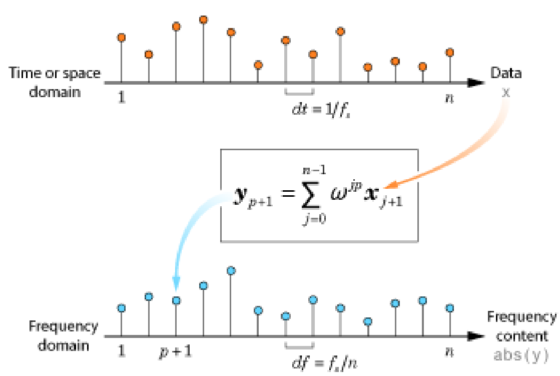


Figura 3: Transformada de fourier

En palabras simples, la Transformada de Fourier pasa del dominio del tiempo, al dominio de las frecuencias, para identificar las frecuencias más importantes en la señal original. Esto es de utilidad, ya que por ejemplo, al ver las frecuencias que representan sólo ruido dentro de la señal, pueden suprimirse y dejar la señal libre de este ruido.

# 3. Desarrollo de la experiencia

El desarrollo del laboratorio se llevó a cabo en 4 fases o iteraciones relevantes, dado esto se mostrará toda la información relevante a cada una de estas partes con su respectiva explicación.

** **Nota:** *se adjuntan sólo los códigos correspondientes a las operaciones solicitadas, en lo que respecta a los gráficos, no se adjuntó código puesto que todos se generaron bajo el mismo patrón, cambiando solo los parámetros de entrada. Esto con el fin de evitar adjuntar todo el código y que se extensamente largo el informe, ya que, el código se encuentra comentado y explicado en su totalidad.*

## 3.1. $1^a$ Iteración

- **Leer una señal de audio (.wav) y determinar a qué corresponde cada uno de los parámetros retornados.**
- **Graficar la función de audio en el tiempo**

En esta Iteración, se realizó la lectura del archivo 'Handel.wav', separándolo en sus respectivas componentes, mediante el uso de la librería Scipy y su función para archivos .wav, obteniendo el siguiente código en Python:

```
# f -> frecuencia de handel.wav
# audio -> arreglo de amplitudes de handel.wav y tipo de dato
f,audio = wavfile.read("handel.wav")

# Numero de amplitudes dentro de Handel.wav
amplitudes = len(audio)

# Arreglo con valores de tiempo para cada amplitud
Time = np.linspace(0, amplitudes/f, amplitudes)
```

Cabe destacar que las componentes del audio leído son:

- **f:** Frecuencia de muestreo del audio. Es un número entero cuyo valor es 8192.

- **audio:** Arreglo de dos elementos, el primero, una lista de enteros que representan las amplitudes del audio,el segundo, el tipo de dato del archivo (int16).

Además mediante el uso de la librería matplotlib, se graficó el audio en función del tiempo, obteniendo así el siguiente gráfico:

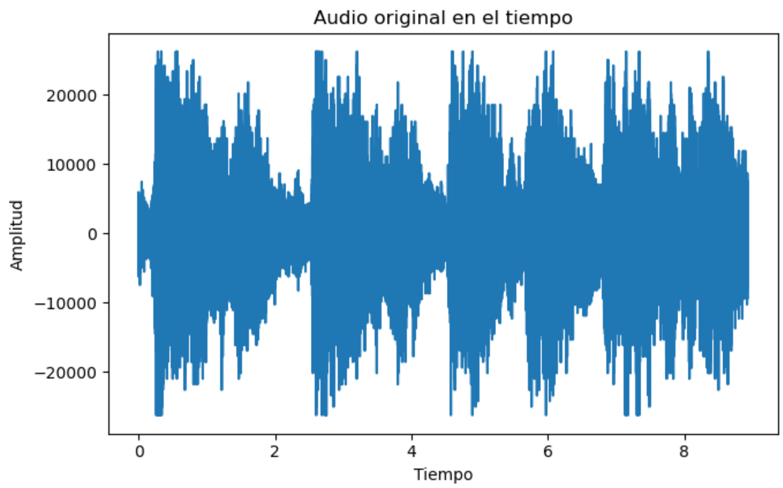


Figura 4: Gráfico del audio original

## 3.2. $2^a$ Iteración

- **Calcule la transformada de Fourier de la señal de audio.**
- **Grafique su señal en el dominio de la frecuencia.**
- **Calcule su transformada de Fourier inversa.**
- **Grafique su señal en el dominio de la frecuencia.**

Para esta iteración, se calculó la transformada de fourier del audio, luego se obtuvo su gráfico, se calculó su transformada inversa y finalmente se graficó dicha inversa con el fin de facilitar la comparación.

```
# Se obtiene la Transformada de Fourier Normalizada!
fourierY = fft(audio) / amplitudes

```

```
# Se obtiene la frecuencia de cada punto de la Transformada de Fourier
fourierX= fftfreq(int(amplitudes), 1.0/f)
```

Para la transformada de fourier se utilizó nuevamente la librería Scipy, específicamente las herramientas fft (Fast Fourier Transform) y fftfreq (Fast Fourier Transform Frequencies), las cuales entregaron los resultados buscados e implicitamente, ambos ejes necesarios para realizar el gráfico que se adjunta a continuación:

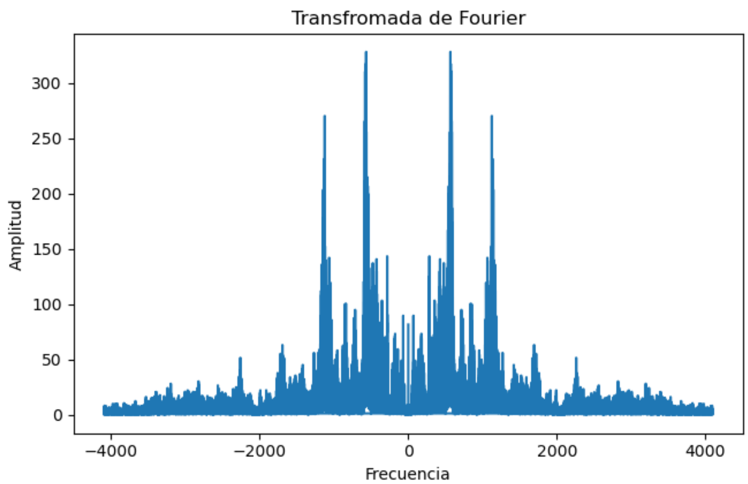


Figura 5: Gráfico de la transformada de fourier

Para efectos de esta experiencia, se optó por trabajar con la parte absoluta de la transformada de fourier, puesto que ésta es una función continua, lo que significa que solo debe poseer 1 valor de y por cada x.

```
# Se obtiene la transfromada de fourier inversa y normaliza
fourierInversaY = ifft(fourierY).real * amplitudes
...
plt.plot(fourierX,abs(fourierY))
```

Una vez que se calcula la transformada de fourier y se grafica su parte absoluta, se obtiene como resultado un gráfico ( que se adjunta a continuación) idéntico al Gráfico 1.

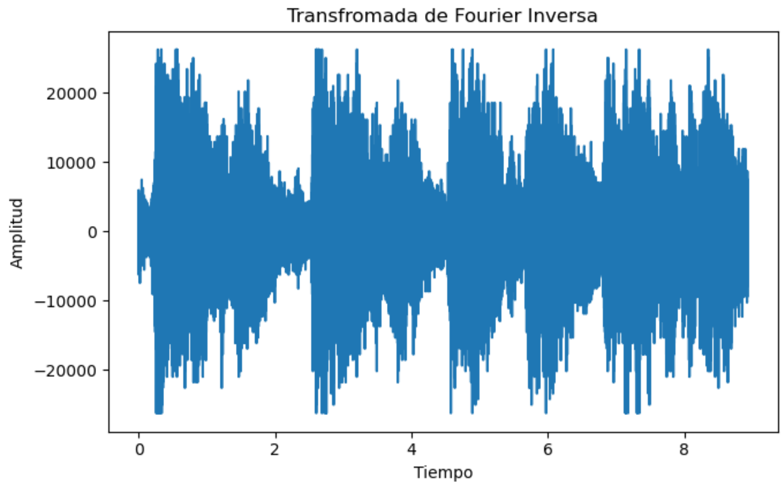


Figura 6: Gráfico de la transformada de fourier inversa

### 3.3. $3^a$ Iteración

- **Calcule y grafique el espectrograma de la función**

Para esta iteración se calculó y graficó el espectograma del audio original, para esto se utilizó la función *specgram* que realiza ambas operaciones al instante dado un audio y frecuencia determinada.

```
# Calculo y grafico del espectograma
plt.specgram(audio,Fs=f)
```

Obteniendo como resultado, el gráfico en 3 dimensiones del espectograma que se adjunta a continuación:

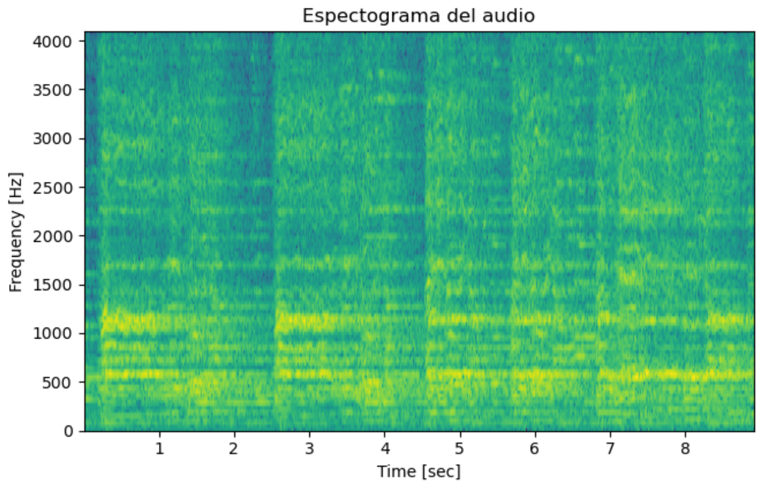


Figura 7: Espectograma de la señal original

### 3.4. $4^a$ Iteración

- **Filtre el ruido de la señal de audio original.**
- **Diseñe un filtro FIR para eliminar el ruido de la señal de audio.**
- **Obtenga la transformada de Fourier y el espectrograma de la señal filtrada.**

En esta última iteración, se crea el filtro solicitado con una configuración de filtro tipo pasabanda, para ello se utilizó nuevamente la librería Scipy, específicamente la función *butter* del segmento *signal*. Luego con la función lfilter se aplicó el filtro al audio original. Cabe destacar que el rango elegido fue probado con distintos valores pero se dejó ese pues en internet recomiendan esa configuración para eliminar ruidos leves (o de bajas frecuencias) en los audios.

```
# Se define el rango en el que efectuara el filtro
ini = 300
end = 1200

# Se crea el filtro, de orden 6, aplicando las condiciones de un tipo de
    rango de pasa banda
b, a = butter(N=6, Wn=[2 * ini/f, 2 * end/f], btype='band')

```

```
# Se filtra el audio
afiltrado = lfilter(b, a, audio)

# Se guarda el nuevo archivo de audio filtrado
wavfile.write("AudioFiltrado.wav", f, afiltrado.astype(np.int16)) #dtype=
    np.int16
```

Luego de esta implementación, se obtuvo el siguiente gráfico de función en el tiempo que, a simple vista, es muy similar al audio original, no obstante posee diferencias con éste ya que es posible apreciar que varias amplitudes son mucho más delgadas (respecto al eje x) que las del audio original.

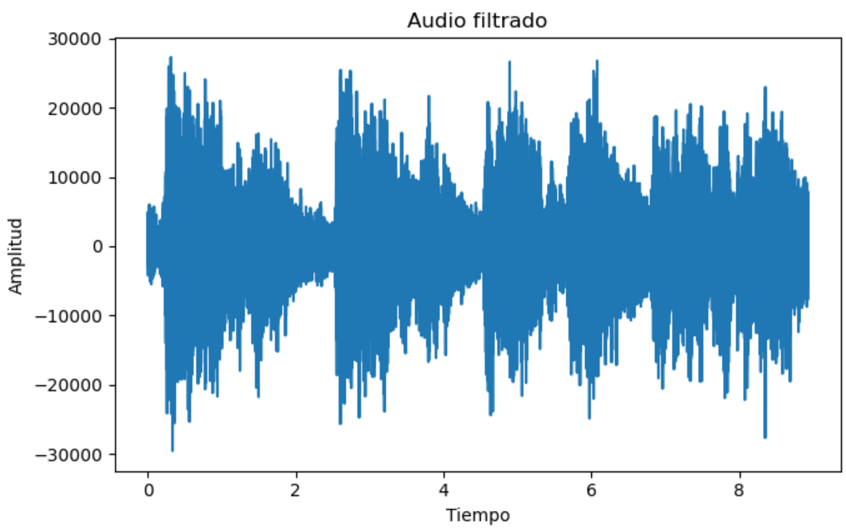


Figura 8: Gráfico de la señal filtrada

Luego, se utilizó el mismo procedimiento que en la iteración 2 para calcular y graficar la transformada de fourier.

```
# Obtencion de la Tranformada de Fourier Normalizada!
fourierY = fft(afiltrado)/amplitudes
```

A continuación se adjunta el grafico obtenido, como se es posible ver, las diferencias son mucho más notorias que en gráfico anterior.

Finalmente, también de la misma forma que en la iteración anterior, se obtuvo el espectograma del nuevo audio filtrado.

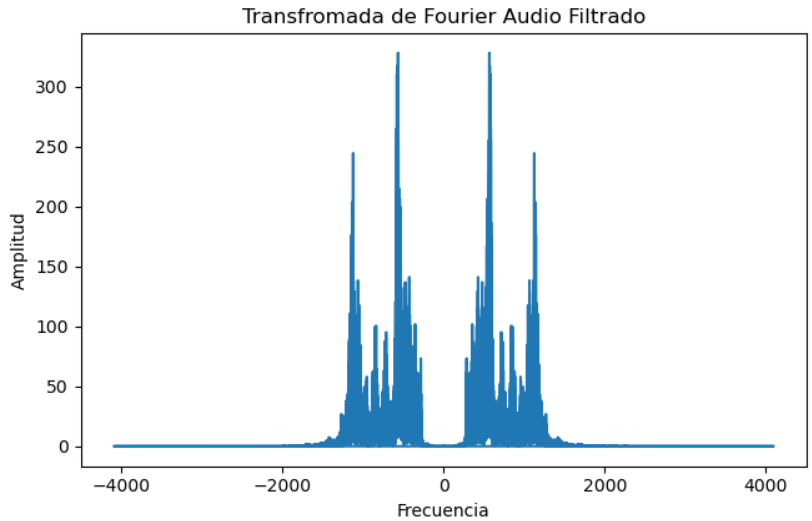


Figura 9: Gráfico de la transformada de fourier filtrada

```
plt.specgram(afiltrado,Fs=f)
```

El espectrograma que se obtuvo posee también a simple vista notables diferencias con el espectograma del audio original.

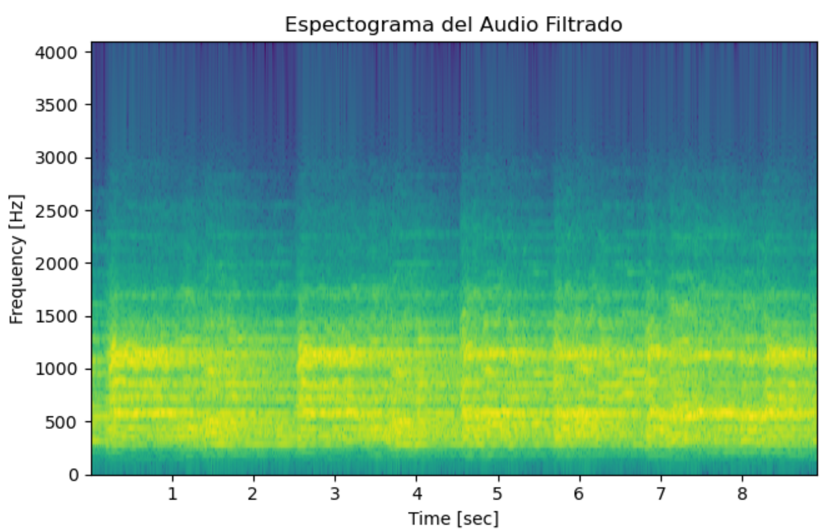


Figura 10: Espectograma de la señal filtrada

# 4. Análisis de resultados

A continuación se presentará el análisis de los resultados obtenidos en cada iteración de la experiencia.

## 4.1. $1^a$ Iteración

Entre los resultados obtenidos para esta iteración, se puede ver que para el primer gráfico (Ver Figura 4.) rango de las amplitudes oscilan entre los valores 25,000 y $-25{,}000$ aproximadamente, lo anterior hace sentido ya que el formato .WAV de 16-bit, posee variaciones de amplitudes en su jerarquía de rangos que oscilan entre -32768 y 32768 (Overflow).

## 4.2. $2^a$ Iteración

Para esta iteración, es posible apreciar que las frecuencias mas relevantes en el gráfico de la transformada de fourier (Ver Figura 5.) oscilan entre los valores 300 - 1200 Hz, esto nos permite deducir que tales frecuencias se posicionan en los tonos medios de la escala musical, lo más probable, es que en el de las octavas quinta, sexta y séptima (250 - 1400 Hz), el cual usualmente es rango que abarca a una gran parte de armónicos en la mayoría de las fuentes sonoras. A su vez, las frecuencias con menor amplitud se posicionan en los tonos agudos de las octavas (2000 - 16000 Hz) y algunos graves (16 Hz - 250 Hz).

Es posible también ver que no existen amplitudes de frecuencias sobre los 4000Hz o bajo los -4000Hz, esto hace sentido ya que tales sonidos corresponden a agudos de las últimas octavas, las cuales no están presentes en el audio correspondiente.

Por último, es fácil notar que en el gráfico de la transformada de fourier inversa (Ver Figura 6.) coinciden todos los puntos con el gráfico de la señal original de audio (Ver Figura 4.), esto indica que el procedimiento que se llevó a cabo fue el correcto puesto que cumple con las propiedades de la transformada de fourier.

## 4.3. $3^a$ y $4^a$ Iteración

En esta iteración, podemos observar que tal como se mencionó en el desarrollo, el gráfico del audio filtrado (Ver Figura 8.) presenta una leve diferencia, la cual consiste en que el gráfico perdió volumen, esto se debe a que se filtró todas las frecuencias que estaban en el rango determinado en el código que corresponde a 300 - 1200 Hz. Esto es mucho mas perceptible cuando comparamos los gráficos de las transformadas de fourier (Ver Figura 9.) ya que en él se puede ver como el filtro pasa banda solo permite pasar por (valga la redundancia) la banda de paso sólo las frecuencias de tonos medios (tonos medios dentro de los 300 - 1200 Hz ya que recordando, el tono medio está entre 250 - 1400 Hz, pero en esta experiencia se acotó aún más con la intención de que fuera más visible el efecto), limitando todas las frecuencias que anteriormente habíamos catalogado de agudas y graves que no pertenecen al rango mencionado.

Por último, es también destacable la transformación que sufre el espectograma al ser sometido por el filtro de pasabanda (Ver Figura 10.) el que al ser comparado con el espectograma original es posible concluir 2 cosas:

En primer lugar, toda frecuencia mayor a 1200 y menor a 300 Hz ya no mantiene la misma intensidad en el gráfico, casi como si lo hubieran compactado entre esas frecuencias. Esto se debe a que la representación de la energía expresada en dB como el módulo de la amplitud de la Transformada de Fourier ahora se concentra únicamente en las frecuencias que atravesaron la banda de paso, mientras que a medida que se aleja de este rango de frec. la energía disminuye considerablemente.

En segundo lugar y en complemento de lo anterior, para frecuencias mayores a los 3000Hz y cercanos a los 0Hz no existe energía alguna, puesto que al pasar por el filtro, fue suprimida.

De lo anterior podemos destacar que los filtros pasabanda son una herramienta perfecta a la hora de reducir el ruido en las señales puesto que, el ruido se puede considerar como un set de frecuencias agudas y graves que representan un porcentaje minoritario en una señal, por lo tanto, a la hora de usar una banda de paso, podemos literalmente suprimir todas aquellas frecuencias que no queremos dentro de nuestra señal.

# 5. Conclusión

La experiencia de laboratorio número 1 permitió reforzar todos los conocimientos vistos en cátedra acerca de los usos de la transformada de fourier y su inversa. Todo esto gracias a la aplicación y visualización a través de gráficos que sirvieron incluso para demostrar propiedades teóricas como que la inversa de la transformada de fourier de una señal es la señal.

Además permitió entender cómo opera un filtro de pasabanda sobre un audio y el principal uso que se le puede otorgar para eliminar ruidos en señales aproximando a cero las frecuencias mas agudas y graves presentes en ellas.

Finamente cabe destacar que el uso de audios .wav en la experiencia facilitó la obtención de mejores resultados, esto porque al ser un medio multimedia utilizado, se está llevando a la praxis métodos reales con cosas reales, y no una simple simulación.

Como conclusión, los objetivos planteados fueron cubiertos a cabalidad, ya que a través del uso de las herramientas "Transformada de Fourier", "Transformada Inversa", "Filtrosz .Espectogram"se abarcaron todos los tópicos básicos necesarios para la comprensión de las señales junto con Fourier y sus usos y aplicaciones.

# Bibliografía


[Overflow] Overflow, S. Wav properties. https://stackoverflow.com/questions/59658218/please-explain-why-we-use-32768-in-audio-signals.

[Rosario] Rosario, U. N. Electrónica 3. www.fceia.unr.edu.ar/enica3/da-ad.pdf.

[Tanenbaum] Tanenbaum. *Redes de Computadoras*.