

# IMT2220 Cálculo para Ciencia de Datos - 2025-2

## Tarea 3

Profesor: Ignacio Labarca ([ignacio.labarca@uc.cl](mailto:ignacio.labarca@uc.cl))

Fecha de entrega: 10 de diciembre de 2025 (por Canvas)

### 1. Introducción

En esta tarea estudiaremos un primer acercamiento al análisis de señales de audio mediante el uso de la transformada de Fourier. El análisis de Fourier es una herramienta fundamental en el procesamiento de señales que permite descomponer una señal temporal en sus componentes de frecuencia, revelando información que no es evidente en el dominio del tiempo.

Trabajaremos con dos pistas de audio distintas, lo que nos permitirá observar diferentes comportamientos espectrales y comprender cómo las características de una señal se reflejan en su representación en frecuencia. A través de ejercicios prácticos, aprenderemos a cargar, visualizar, analizar y manipular señales de audio usando Python.

Para la implementación, utilizaremos las siguientes herramientas de Python:

- `scipy.io.wavfile` para leer y escribir archivos de audio en formato WAV.
- `numpy` y sus funciones `rfft` e `irfft` para el cálculo de la transformada de Fourier (dado que trabajamos con señales reales).
- `matplotlib` para la generación de gráficos.

La tarea se entregará en formato Jupyter Notebook.

### 2. Tarea

#### 2.1. Parte 1

En esta parte trabajaremos con la pista de audio `piano-sample.wav`.

- (a) Cargar el archivo e identificar: total de muestras, la tasa de muestreo (*sample rate*) y la duración del audio. Lea la documentación de `wavfile.read` para facilitar los cálculos.
- (b) La pista de audio está en formato estéreo, es decir, hay valores para canales izquierdo y derecho. Para efectos de la tarea, podemos tomar la primera componente y trabajar exclusivamente en ella. Extraiga esta componente del arreglo y normalice los datos para que se encuentren en el intervalo  $[-1, 1]$ .

- (c) Grafique la amplitud de la señal de audio a lo largo del tiempo: para ello, use el arreglo normalizado y genere un intervalo de tiempo considerando la tasa de muestreo calculada. Comente el gráfico obtenido, teniendo en cuenta lo que escucha en la pista de audio.
- (d) Calcule la FFT (con `np.fft.rfft`) y las frecuencias medidas en Hertz (con `np.fft.rfftfreq`). Obtenga un arreglo con la magnitud (valor absoluto) asociada a cada frecuencia.
- (e) Genere el gráfico de espectro de frecuencias: magnitud vs frecuencia. ¿Qué observa en el gráfico? Identifique las 10 frecuencias con mayor magnitud.
- (f) Investigue a qué notas musicales corresponden estas frecuencias en el contexto de un piano.
- (g) Escriba dos filtros en el dominio de frecuencias: el primero establece en cero todo lo que sea mayor a una frecuencia de 400 Hertz. El segundo remueve las 3 frecuencias con valores más altos ( $\pm 10$  Hertz). Copie los datos (con `.copy()`) en nuevos arreglos y aplique cada filtro. Tome la transformada inversa (con `np.fft.irfft`) de cada uno de los arreglos filtrados, usando el parámetro `n igual a la longitud original del audio`.
- (h) Reescala los datos multiplicando por 32767. Guarde dos nuevas pistas de audio para cada uno de los filtros del punto anterior con `wavfile.write`. Describa cómo se compara cada pista filtrada comparada con la original.

## 2.2. Parte 2

En esta parte trabajaremos con la pista de audio `arnold-sample.wav`.

- (a) Cargar el archivo e identificar: total de muestras, la tasa de muestreo y la duración del audio.
- (b) La pista de audio está en formato estéreo, es decir, hay valores para canales izquierdo y derecho. Para efectos de la tarea, podemos tomar la primera componente y trabajar exclusivamente en ella. Extraiga esta componente del arreglo y normalice los datos para que se encuentren en el intervalo  $[-1, 1]$ .
- (c) Grafique la amplitud de la señal de audio a lo largo del tiempo. Comente el gráfico obtenido, teniendo en cuenta lo que escucha en la pista de audio.
- (d) Calcule la FFT (con `np.fft.rfft`) y las frecuencias medidas en Hertz (con `np.fft.rfftfreq`). Obtenga un arreglo con la magnitud (valor absoluto) asociada a cada frecuencia.
- (e) Genere el gráfico de espectro de frecuencias. ¿Qué observa en el gráfico? Identifique las 10 frecuencias con mayor magnitud. A diferencia de la Parte 1, ahora se observan muchos más *peaks* a lo largo del espectro de frecuencia. Investigue acerca de los armónicos y la frecuencia fundamental. Determine si se trata de ese fenómeno y para qué frecuencias en particular.
- (f) Escriba cuatro filtros en el dominio de frecuencias: (i) frecuencias mayores a 1000 Hz, (ii) frecuencias menores a 1000 Hz, (iii) magnitud mayor a 1000, (iv) magnitud menor a 1000. Copie los datos (con `.copy()`) en nuevos arreglos y aplique cada filtro. Tome la transformada inversa (con `np.fft.irfft`) de cada uno de los arreglos filtrados, usando el parámetro `n igual a la longitud original del audio`.
- (g) Guarde las pistas de audio para cada uno de los filtros del punto anterior con `wavfile.write`. Describa cómo se compara cada pista filtrada comparada con la original.

### **2.3. Parte 3**

En esta parte continuaremos trabajando con la pista de audio `arnold-sample.wav`.

- (a) Repita los pasos (a) a (e) de la Parte 2, pero ahora trabajando únicamente con los primeros segundos de audio donde solo hay percusión.
- (b) ¿Qué frecuencias representan el sonido de campanas y percusión al comienzo de la canción? Escriba una estrategia para intentar removerlos del audio completo.

### **3. Formato de entrega**

La tarea debe entregarse a través de Canvas en un archivo comprimido `.zip` que contenga:

- Un Jupyter Notebook con todo el desarrollo de la tarea, incluyendo código, gráficos y comentarios para cada parte.
- Todas las pistas de audio generadas en cada punto de las Partes 1, 2 y 3.

Asegúrese de que el Jupyter Notebook sea autocontenido y pueda ejecutarse correctamente con las pistas de audio originales (`piano-sample.wav` y `arnold-sample.wav`).