

Tareas 3 y 4: Introducción al análisis y procesamiento de audio con Python

Juan M. Haut¹ Andres J. Sanchez-Fernandez²

¹Departamento de Tecnología Computadores y de las Comunicaciones
Área de Arquitectura y Tecnología de Computadores
Universidad de Extremadura
juanmariahaut@unex.es

²Universidad de Extremadura
sfandres@unex.es

Curso 2023–2024



Consideraciones previas

Consejo

Empezar en un repositorio nuevo y ser ordenado. Eliminar todos los entornos de conda que esten en el equipo menos audiopy y base (lo puedes hacer ya) y también aquellos instalados como kernels (tras haber realizado el Punto 1.4 de la siguiente diapositiva). Los comandos son:

```
1 conda env list
2 conda remove --all --name=<env_name>
```

```
1 jupyter kernelspec list
2 jupyter kernelspec uninstall <env_name>
```

donde <env_name> es el nombre del entorno/kernel a eliminar.



Ejercicio de clase: Tarea 3 parte I

1 Configurar el repositorio de GitHub, conda y JupyterLab:

- 1 Crear un repositorio vacío de GitHub para la tarea y clonarlo en local.
- 2 Crear un entorno de conda para trabajar con el repositorio.
Si lo has hecho bien deberá aparecer el nombre si ejecutas `conda env list`.
- 3 Añadir la carpeta `.ipynb_checkpoints/` al `.gitignore` para ignorarla.
Si no lo haces seguramente provocarás un fallo grave cuando envíes a remoto.
- 4 Activar el entorno conda creado.
- 5 Instalar Python 3.10, ipykernel y JupyterLab en el entorno.
- 6 Añadir el entorno conda a los kernels de JupyterLab.
- 7 Instalar JupyterLab en el entorno conda.
- 8 Ejecutar JupyterLab y crear un notebook vacío, seleccionando como kernel el entorno de conda creado anteriormente.
Si no te aparece, la instalación no se ha realizado correctamente.
- 9 Juega con el notebook para familiarizarte con los tres tipos de celda: Code, Markdown, y Raw. Prueba a ejecutar códigos sencillos de Python.

Ejercicio de clase: Tarea 3 parte II

- ② **Análisis de audio con Python y JupyterLab:** Incluye los siguientes puntos en un notebook—con explicaciones detalladas y sin copiar—respetando el mismo orden y utilizando secciones para organizar el contenido:
- ① Organiza todo el contenido del notebook en secciones utilizando celdas de tipo Markdown (#, ##, ###). Cada sección debe estar comentada en detalle y ser visualmente atractiva.
 - ② Comenta todo el código que vayas desarrollando a la derecha de cada línea. No uses IA generativa; se busca que comprendas todo lo que haces.
 - ③ Cargar un audio estéreo de entre los facilitados (no todos que sino el repositorio pesará mucho) y mostrar sus características: frecuencia de muestreo, número de canales y tamaño del archivo en Mb.
 - ④ Incluir un widget para reproducir el audio estéreo.
 - ⑤ Convertir el archivo de audio estéreo a mono y mostrar las características anteriormente mencionadas.
 - ⑥ Incluir un nuevo widget para reproducir el audio mono.
 - ⑦ Mostrar la gráfica en el dominio del tiempo para el audio mono y estéreo (investiga sobre la frecuencia de muestreo).
 - ⑧ Explicar con tus palabras: diferencia entre audio estéreo y mono.



Ejercicio de clase: Tarea 3 parte III

3 Comprobar que no haya fallos.

- 1 Hacer click en *Kernel* → *Restart Kernel and Run All Cells* para reiniciar el kernel y comprobar que no haya fallos.
 - 2 Hacer click en *Kernel* → *Restart Kernel and Clear All Outputs*. Hacemos esto para reducir el tamaño del fichero (evitando el límite de GitHub de 100 Mb) y poder sincronizarlo sin problemas. Guardar el notebook de Jupyter.
 - 3 Exportar los paquetes instalados en tu entorno conda creando el fichero `environment.yml`.
 - 4 Hacer público el repositorio de GitHub (si no está hecho ya).
- 4 Sincroniza todo con GitHub.
- 5 Escribir la url del repositorio en el cuadro de texto de la tarea habilitada en el campus virtual. No hagáis más cambios en el repositorio.

Ejercicio de clase: Tarea 3 parte IV

Fin

Ejercicio de clase: Tarea 4 parte I

❶ **Procesamiento de audio con Python y JupyterLab:** continuamos sobre la práctica de la semana pasada. Utiliza el mismo repositorio que creaste ya.

- ❶ Lee la teoría incluida en el nuevo repositorio: *uex-audiopy2*.
- ❷ Mostrar la gráfica en el dominio del tiempo para el audio mono y estéreo.
- ❸ Explicar con tus palabras: frecuencia de muestreo, *aliasing*, profundidad de bits, ancho de banda y tasa de bits.
- ❹ → Audio mono. Aplicar la Transformada rápida de Fourier (FFT) para cambiar al dominio de la frecuencia. Mostrar la gráfica y explicar el por qué.
- ❺ Calcular la energía del espectrograma y la frecuencia de corte eligiendo un epsilon (probad diferentes pero comentad solo uno).
- ❻ Comprimir la onda aplicando *downsampling*, donde el factor se obtiene a partir de la frecuencia de corte anteriormente calculada.
- ❼ Mostrar el espectrograma de ambas ondas: original y comprimida. Explicar las diferencias.
- ❽ Mostrar el tamaño de ambos archivos en Mb.
- ❾ Colocar también dos widgets para reproducir los audios—original y comprimido—y apreciar las diferencias en calidad.



Ejercicio de clase: Tarea 4 parte II

3 Comprobar que no haya fallos.

- 1 Hacer click en *Kernel* → *Restart Kernel and Run All Cells* para reiniciar el kernel y comprobar que no haya fallos.
 - 2 Hacer click en *Kernel* → *Restart Kernel and Clear All Outputs*. Hacemos esto para reducir el tamaño del fichero (evitando el límite de GitHub de 100 Mb) y poder sincronizarlo sin problemas. Guardar el notebook de Jupyter.
 - 3 Exportar los paquetes instalados en tu entorno conda creando el fichero `environment.yml`.
 - 4 Hacer público el repositorio de GitHub (si no está hecho ya).
- 4 Sincroniza todo con GitHub.
- 5 Escribir la url del repositorio en el cuadro de texto de la tarea habilitada en el campus virtual. No hagáis más cambios en el repositorio.

Ejercicio de clase: Tarea 4 parte III

Fin