

# Tarea 3

**CC5213 – Recuperación de Información Multimedia**

**Profesor: Juan Manuel Barrios**

**Fecha: 30 de octubre de 2020**

El objetivo de esta tarea es implementar un detector de avisos comerciales que solo considera el audio de la señal de televisión. Dada una carpeta que contiene audios con la emisión de un canal de televisión y una carpeta con audios de avisos comerciales usted deberá implementar un software que determine cada una de las emisiones de los comerciales buscados.

El resultado de su tarea es un archivo de texto con todas las ocurrencias de los comerciales detectados en los videos de televisión. Cada ocurrencia debe incluir un valor de confianza donde un valor más alto significa que es más seguro que la detección sea correcta. El archivo debe tener formato de **cinco** columnas separadas por un tabulador (\t), cada aparición en una línea siguiendo el siguiente formato:

```
audio_tv \t tiempo_inicio \t largo \t audio_comercial \t confianza
```

Los tiempos se miden en segundos. Un posible archivo de salida es el siguiente:

televisión.mp3	87.3	30.1	comercial8.mp3	0.2943
televisión.mp3	420.8	15.6	comercial3.mp3	7.0121
televisión.mp3	1892.8	25.9	comercial2.mp3	5.0315
televisión.mp3	2087.5	30.1	comercial3.mp3	2.3875

La tarea se debe dividir en los siguientes tres módulos.

## Módulo 1: Extracción de características

```
python tarea3-descriptores.py audios_tv/ descriptores_tv/
```

```
python tarea3-descriptores.py audios_comerciales/ descriptores_comerciales/
```

Lee todos los archivos con extensión .mp3 que están en la primera carpeta, calcula descriptores de contenido y escribe uno o más archivos en la segunda carpeta.

## Módulo 2: Búsqueda por similitud

```
python tarea3-busqueda.py descriptores_tv/ descriptores_comerciales/ similares.txt
```

Lee los descriptores de televisión y comerciales y escribe en el archivo de salida una lista con los segmentos más cercanos entre televisión y comerciales.

## Módulo 3: Detección de comerciales

```
python tarea3-deteccion.py similares.txt detecciones.txt
```

Lee la lista de segmentos similares del módulo anterior y escribe en el archivo de salida todas las apariciones de los comerciales junto con un valor de confianza.

## Resultados de detección

Su tarea debe reportar la mayor cantidad de comerciales emitidos, con la mejor exactitud posible de los tiempos de inicio y largo, junto con la menor cantidad de detecciones incorrectas.

Junto con este enunciado encontrará conjuntos de prueba conteniendo videos de televisión, videos de comerciales y la respuesta con las detecciones esperadas.

Se publicará un programa llamado `tarea3-evaluar.py` que ejecuta los tres módulos de su tarea y evalúa el resultado de la detección. Cada detección reportada se considera correcta si intersecta una emisión real.

Su tarea será evaluada en los datasets publicados y en otros conjuntos similares. **Su tarea no puede demorar más de 4 horas en total.**

## Entrega

Deberá entregar los códigos fuentes de su tarea y un archivo `Readme.txt` que explica brevemente su implementación, el o los descriptores usados, librerías usadas, forma de compilación y resultados obtenidos.

El plazo máximo de entrega es el **viernes 13 de noviembre** hasta las 23:59 por U-Cursos. No se aceptarán tareas atrasadas.

La tarea puede ser implementada en **Python 3**, **C++ 11** o **Java 8**. Puede utilizar cualquier librería gratuita (por ejemplo LibRosa, NumPy, SciPy u otra).

Debe enviar solo el código fuente de su tarea (archivos `.cpp`, `.py` o `.java`) y el archivo `Readme.txt`. No enviar descriptores calculados.

## Evaluación

Se evaluarán tres aspectos: orden del código fuente (20%), claridad del archivo `Readme.txt` (20%) y la calidad de los resultados logrados (60%).

La calidad de los resultados logrados se medirá usando `tarea3-evaluar.py` sobre distintos datasets.

**La tarea es \*individual\*.**