

# Archivos Binarios

## Ejercicio 1


Es sabido que la federalidad y democracia reinan en la cátedra y uno, solo uno de los ámbitos donde se ve reflejada la diversidad es en la música. El departamento de Recursos Humanos armó una bella playlist con varias canciones que cada colaborador seleccionó a su gusto y piacere, sin limitaciones, sin juzgar, sin preguntar.

Se nos disponibilizaron 2 archivos, uno binario de acceso secuencial, llamado **canciones.dat** con las canciones elegidas por los integrantes de la cátedra, compuesto de registros del tipo **cancion\_t** y otro, también binario de acceso directo llamado **generos.dat** donde se guardaron todos los géneros musicales, con la particularidad de que este archivo contiene **ids** de género únicos y correlativos, empezando desde el 1.

```
typedef struct cancion {
    char nombre[MAX_CANCION];
    char artista[MAX_ARTISTA];
    char colaborador[MAX_COLAB];
    int duracion;
    int id_genero;
} cancion_t;
```

```
typedef struct genero {
    char nombre[MAX_GENERO];
    int id;
} genero_t;
```

1. Crear un procedimiento que reciba cómo argumento el nombre de un colaborador y muestre por pantalla las canciones elegidas por ese colab.
2. Crear un procedimiento que reciba cómo argumento el nombre de un colaborador y un número y elimine del archivo de canciones, todas aquellas cuya duración sea menor al número ingresado y pertenezcan a ese colaborador.

- 
3. Crear un procedimiento que reciba como argumento el nombre de un colaborador y muestre por pantalla las canciones elegidas por ese colab y el género al que pertenece cada una.
  4. Sabiendo que el archivo **generos.dat** no contiene más de 20 géneros, ordénelo por nombre.