



TMDB (The Movie Database) es una base de datos en línea que contiene una amplia colección de información sobre películas, series de televisión y programas de streaming. En este examen trabajaremos con dos ficheros distintos relacionados con películas registradas en la base de datos TMDB:

- **“movies\_fp.csv”**: Cada línea contiene información sobre una película separada por comas. Esta información incluye campos como el identificador único de la película (string), el título de la película (string), el idioma original en el que fue producida (string), la fecha de lanzamiento (date), la calificación promedio otorgada por los usuarios (float), su popularidad en la base de datos (int) y una indicación de si es apta sólo para adultos (boolean). Contenido del archivo “movies\_fp.csv”:

```
id,title,original_language,release_date,vote_average,popularity,adult
238,The Godfather,en,1972-03-14,8.7,114574,False ...
```

- **“movies\_fp\_genres.csv”**: Cada línea del fichero contiene un identificador (string) de película y un género al que pertenece la película con ese identificador. Si la película tiene más de un género, su código aparecerá varias veces en el fichero. Por ejemplo, en el siguiente extracto del archivo la película con id 238 tiene los géneros “Drama” y “Fake”. Contenido del archivo “movies\_fp\_genres.csv”:

```
id:genres 238:Drama
238:Fake ...
```

Plantee un proyecto T90\_TMDB\_movie con sus respectivas carpetas **src**, **data** y **doc**. Copie este enunciado en la carpeta **doc** y los ficheros en la carpeta **data**

### **Ejercicio 1: Leer los archivos de películas de TMDB y generar una lista de tuplas con nombre Película (2 puntos)**

Utiliza la librería csv de Python para leer el archivo y generar una lista de namedtuple con la siguiente estructura:

```
Pelicula = namedtuple('Película', ['id', 'title', 'original_language',  
'release_date', 'vote_average', 'popularity', 'adult', 'genres'])
```

Para cada línea en el archivo, deberás crear una Película con los siguientes campos:

- **id**: Una cadena con el identificador único de la película.
- **title**: Una cadena con el título de la película.
- **original\_language**: Una cadena con el idioma original en el que fue producida la película.
- **release\_date**: Un objeto de tipo fecha con la fecha de lanzamiento de la película.
- **vote\_average**: Un valor real con la calificación promedio otorgada por los usuarios.
- **popularity**: Un valor entero con la popularidad de la película en la base de datos.
- **adult**: Un valor booleano que indica si la película es apta para adultos (True o False).

- **genres**: Un conjunto (set) que contiene los géneros a los que pertenece la película.

Escribe una función `Leer_películas` que reciba dos nombres de ficheros (**`nombre_archivo_películas`**, **`nombre_archivo_generos`**) y genere un listado de Película fusionando la información de ambos ficheros.

Recuerde que los nombres de los géneros deben guardarse como conjuntos (sets) para evitar duplicados y que están en un fichero auxiliar (de ahí que la función de lectura de películas reciba dos ficheros).

Para que el código sea más modular implementa una función auxiliar `Leer_diccionario_generos` que reciba el nombre un fichero, con una estructura similar a la del fichero de géneros, y devuelva un diccionario en el que la clave es el id de una película y el valor es un conjunto con los géneros de esa película. Así para el fichero proporcionando, el diccionario resultado de la función debería ser {238:{"Drama", "Fake"}, ...}. A medida que vayas leyendo la información de cada película del fichero principal, puedes acceder al diccionario para completar con la información auxiliar de dicha película.

### **Ejercicio 2: Obtén el género más frecuente en el listado de películas (1.5 puntos)**

En este ejercicio, tu tarea es implementar una función **`genero_mas_frecuente`** que tome una lista de tuplas de tipo Película y genere una tupla con el nombre del género más frecuente y la frecuencia.

### **Ejercicio 3: Obtén un diccionario de películas mejor valoradas por idioma (2 puntos)**

En este ejercicio, tu tarea es crear la función **`mejor_valorada_por_idioma`** que tome una lista de tuplas de tipo Película y genere un diccionario en el que las claves sean los idiomas y el valor asociado a cada clave será la película mejor valorada en el idioma al que hace referencia la clave.

La película mejor valorada en un idioma se define como aquella que tenga la mayor popularidad, y en caso de empate en popularidad, se seleccionará la película con la mejor calificación promedio otorgada por los usuarios.

### **Ejercicio 4: Calcula la media de calificaciones promedio para un conjunto de géneros (1.5 puntos)**

En este ejercicio, tu tarea es crear una función llamada **`media_calificaciones`** que tome como entrada lista de tuplas de tipo Película y un conjunto de géneros, y devuelva la media de las calificaciones promedio (**`vote_average`**) de las películas que contengan todos los géneros dados como parámetro.

### **Ejercicio 5: Obtener el top n de películas por genero (2 puntos)**

Tu tarea en este ejercicio es crear una función llamada **`top_n_por_genero`** que tome como entrada lista de tuplas de tipo Película y un valor entero n, y devuelva un diccionario en el que las claves sean los géneros y el valor asociado a cada clave sea una lista con las n películas de ese género con mayor calificación promedio (**`vote_average`**).

### **Ejercicio 6: Comprobación de Resultados (1 punto)**

Para verificar la correcta implementación de los ejercicios, deberás crear un script de prueba que llame a una función de test para probar las funciones solicitadas en los apartados anteriores y muestre los resultados obtenidos. No es necesario probar las funciones auxiliares.

A continuación, se proporciona **la salida esperada** que puedes utilizar como base para generar los test

-----

#### TESTING EJERCICIO 1

203

Primera: Pelicula(id='238', title='The Godfather', original\_language='en',  
release\_date=datetime.date(1972, 3, 14), vote\_average=8.7, popularity=114574,  
adult=False, genres={'Fake', 'Drama'})

Última: Pelicula(id='347688', title='Scooby-Doo! and KISS: Rock and Roll  
Mystery', original\_language='en', release\_date=datetime.date(2015, 7, 10),  
vote\_average=8.1, popularity=16717, adult=False, genres={'Adventure', 'Drama'})

-----

#### TESTING EJERCICIO 2

El género más frecuente es: ('Adventure', 174)

-----

#### TESTING EJERCICIO 3

Mejor en español (es): Pelicula(id='1010581', title='My Fault',  
original\_language='es', release\_date=datetime.date(2023, 6, 8),  
vote\_average=8.3, popularity=1034594, adult=False, genres={'Action',  
'Adventure', 'Science Fiction'})

-----

#### TESTING EJERCICIO 4

'Action', 'Adventure', 'Fake':

0.0

'Action', 'Thriller':

8.45

-----

#### TESTING EJERCICIO 5

Top 2 Fake: [Pelicula(id='238', title='The Godfather', original\_language='en',  
release\_date=datetime.date(1972, 3, 14), vote\_average=8.7, popularity=114574,  
adult=False, genres={'Fake', 'Drama'})]

Top 2 Action: [Pelicula(id='129', title='Spirited Away',  
original\_language='ja', release\_date=datetime.date(2001, 7, 20),  
vote\_average=8.5, popularity=72546, adult=False, genres={'Action',  
'Adventure'}), Pelicula(id='155', title='The Dark Knight',  
original\_language='en', release\_date=datetime.date(2008, 7, 14),  
vote\_average=8.5, popularity=92305, adult=False, genres={'Thriller', 'Action',  
'Adventure'})]

-----