



En este examen vamos a analizar datos de títulos de producciones audiovisuales que se pueden ver en la plataforma Netflix. Para ello, partiremos de un conjunto de datos de la plataforma Kaggle [1] del que extraeremos el siguiente conjunto de propiedades:

- **Título:** nombre de la producción audiovisual, de tipo String, consultable.
- **Tipo:** una propiedad con dos valores posibles SHOW o MOVIE, de tipo enumerado, consultable.
- **Año de producción:** año en de la producción, de tipo Integer, consultable.
- **Duración:** duración de cada capítulo o película completa si es de tipo MOVIE, de tipo Duration, consultable.
- **Géneros:** conjunto de géneros cinematográficos de la producción, de tipo Set<String>, consultable.
- **Número de temporadas:** total de temporadas de la producción que pueden verse en la plataforma, de tipo Integer, consultable.
- **Score IMDB:** valoración de la producción en la plataforma IMDB [2], de tipo Double, consultable.
- **Popularidad IMDB:** número de votos registrados en la plataforma IMDB, de tipo Long, consultable.

La información de cada producción se obtiene a partir de un fichero CSV en formato UTF-8 cuyas líneas contienen los siguientes datos: título, tipo, año de producción, duración en minutos, géneros, número de temporadas, score en IMDB y popularidad (votos) en IMDB.

A continuación, se muestran algunas líneas de ejemplo del fichero CSV.

```
Five Came Back: The Reference Films,SHOW,1945,48,['documentation'],1,0,0
Taxi Driver,MOVIE,1976,113,['crime'; 'drama'],0,8.3,795222.0
The Guns of Navarone,MOVIE,1961,158,['war'; 'action'; 'drama'],0,7.5,50150.0
```

Cree en el proyecto los paquetes `fp.netflix`, `fp.netflix.test` y `fp.utiles`, y dentro de ellos implemente los tipos que se piden a continuación.

### Ejercicio 1: Tipo base (1 punto)

Implemente el tipo `ProduccionNetflix` mediante un *record*, de acuerdo con la siguiente información:

#### Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo, en el orden en el que están definidas.

#### Restricciones:

- R1: el año de producción debe ser posterior a 1900.
- R2: el score de IMDB debe estar comprendido entre 0 y 10.
- R3: la popularidad en IMDB debe ser igual o superior a 0.
- R4: si el tipo es MOVIE, el número de temporadas debe ser 0. Si el tipo es SHOW, el número de temporadas debe ser mayor o igual a 1.

Criterio de igualdad: dos producciones son iguales si tienen el mismo título y año de producción.

Orden natural: los objetos `ProduccionNetflix` se ordenan alfabéticamente por título y, a igualdad de título, por año de producción.

Representación como cadena: una cadena que contenga los valores de cada propiedad en el orden en el que se han definido en la descripción del tipo.

## Ejercicio 2: Tipo contenedor (0.5 puntos)

Implemente el tipo `CatalogoNetflix` de acuerdo con la siguiente descripción:

Propiedades:

- **producciones:** conjunto de producciones recogidos en el catálogo Netflix ordenados por año de producción, de tipo `SortedSet<ProduccionNetflix>`, consultable. **No debe ser posible añadir o eliminar elementos al conjunto desde fuera del tipo contenedor.**

Constructores:

- C1: recibe un parámetro de tipo `Stream<ProduccionNetflix>` para inicializar el conjunto de producciones.

Criterio de igualdad: dos catálogos son iguales si sus conjuntos de producciones son iguales.

Representación como cadena: muestra el conjunto de producciones del catálogo.

## Ejercicio 3: Factoría (1.5 puntos)

Cree una factoría *FactoriaNetflix* que permita leer los datos de un fichero CSV y almacenarlos en el tipo contenedor. La factoría debe contener los siguientes métodos:

- *CatalogoNetflix leeNetflix(String rutaFichero):* lee un fichero con los datos CSV de los objetos de tipo `ProduccionNetflix` registrados y construye un objeto de tipo `CatalogoNetflix`.
- *ProduccionNetflix parse(String s):* crea un objeto de tipo `ProduccionNetflix` a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.

## Ejercicio 4: Tratamientos secuenciales (7 puntos)

Añada los siguientes tratamientos secuenciales al tipo contenedor. Debe resolver todos los métodos **mediante streams**, salvo que se le indique expresamente lo contrario:

1. *Map<String, Set<ProduccionNetflix>> getProduccionesPorGenero():* devuelve un `Map` que relaciona cada género con el conjunto de producciones de ese género. Resuélvalo sin usar streams. **(1 punto)**
2. *SortedSet<String> getGeneros():* devuelve un conjunto ordenado con todos los géneros. **(1 punto)**
3. *List<String> getTitulosDeGenerosOrdenadosPorTamanyo(List<String> genres):* dada una lista de géneros (genres) devuelve una lista con los títulos cuyos géneros contengan **TODOS** los géneros en genres. La lista deberá estar ordenada de mayor a menor por número de géneros asignados a la producción y, en caso de empate, por tamaño (número de caracteres) del título. **(1 punto)**
4. *String getGeneroConMayorPopularidadAcumulada():* devuelve el género con mayor popularidad acumulada (suma de popularidad IMDB de todas las producciones registradas en el catálogo). **(1.5 puntos)**
5. *SortedMap<Integer, Double> getMediaTopNScoresDeGeneroPorAnyo(String g, Integer n):* dado un género g y un entero n, devuelve un `Map` ordenado que tiene como clave los años de producción y como valor la media de los scores IMDB para las producciones de género g. Dicha media se calculará exclusivamente con las n producciones con un score más alto. **(1.5 puntos)**

Escriba un test para el tipo contenedor. En este test se leerán los datos del fichero CSV y se probarán todos los métodos del ejercicio 4. Cada método del ejercicio 4 tiene que tener su correspondiente método de test (tal como se ha hecho en clase). **(1 punto)**

[1] <https://www.kaggle.com/datasets/victorsoeiro/netflix-tv-shows-and-movies>

[2] <https://www.imdb.com/>