

# CATÁLOGO DE PELÍCULAS

Aldana Smyrna Medina Lostaunau ([aldana.medina@udc.es](mailto:aldana.medina@udc.es))

Claudia Vidal Otero ([claudia.votero@udc.es](mailto:claudia.votero@udc.es))

## **Manual de Usuario: Simulador gestión de catálogo de películas**

Al iniciar este programa, puedes cargar películas desde un archivo de texto, eliminar duplicados, buscar películas por director o año de estreno, y guardar las películas en un archivo ordenado. A continuación, tiene una guía paso a paso para comenzar a utilizar el catálogo:

### 1. Preparación del Archivo de Entrada

Antes de comenzar, asegúrate de tener un archivo de texto que contenga la información sobre las películas que quieres subir como catálogo. El archivo debe seguir el siguiente formato:

**Director; Título; Año; Rating**

Ejemplos:

Meyers, Nancy; The Intern; 2015; 7.1

Kubrick, Stanley; 2001: A Space Odyssey; 1968; 8.3

De Palma, Brian; Phantom of the Paradise; 1974; 7.3

Cada película debe estar en una línea separada y los campos deben estar separados por punto y coma. Además, el rating de cada película solo puede tener los valores de 0 a 10. Una vez que hayas cargado las películas, podrás acceder a todas las funcionalidades de la aplicación.

### 2. Ejecución del Programa

Para ejecutar el programa, abre una terminal y navega hasta el directorio donde se encuentran los archivos del simulador (main.py, movies.py), los archivos necesarios para el manejo de listas posicionales (linked\_ordered\_positional\_list.py, que requiere doubly\_linked\_base.py, o array\_ordered\_positional\_list.py, que puede ser utilizado de forma independiente.) y el archivo de entrada. Luego, ejecuta el programa de la siguiente manera:

```
python main.py
```

A continuación, te saldrá en la terminal el siguiente mensaje:

“Ingrese el nombre del archivo de películas:”

Deberás escribir el nombre de un archivo en este formato: *nombre\_archivo.txt*

### 3. Funcionalidades principales del Menú:

**Mostrar todas las películas:** Esta primera opción muestra todas las películas cargadas en la aplicación, junto con su director, año de estreno y rating. De haber múltiples copias y/o versiones de una película, se incluyen todas en el listado.

**Buscar películas por director:** Puedes buscar películas filtradas por el nombre del director. La aplicación mostrará todas las películas dirigidas por el director especificado, el nombre del director deberá tener el siguiente formato para poder ser leído por la aplicación: “Apellido, Nombre”.

**Buscar películas por año de estreno:** Esta función te permite buscar películas filtradas por el año de estreno. La aplicación mostrará todas las películas lanzadas en el año especificado.

**Generar nuevo archivo con las películas ordenadas y sin duplicados:** Puedes guardar las películas en un archivo que se llamará por defecto "películas\_ordenadas.txt".

**Mostrar métricas:** Esta opción muestra una serie de métricas respecto a los datos del archivo, concretamente el número de películas por director, la puntuación media por director y la puntuación media por año de estreno

**Salir:** Esta opción facilitará tu salida del programa, para volver a iniciarlo deberás volver a iniciar el paso 2: "Ejecución del programa."

### Descripción de las Fases de Desarrollo

#### 1. Análisis de Requisitos:

En esta fase, se identificaron los requisitos del proyecto basados en la especificación proporcionada. Se necesita gestionar un catálogo de películas con ciertas funcionalidades requeridas: la carga de datos desde un archivo, la ordenación y eliminación de duplicados, y la visualización de listados y métricas.

#### 2. Arquitectura del Sistema:

La arquitectura del sistema sigue un enfoque de programación orientada a objetos (OOP). Se utilizan listas posicionales para almacenar y ordenar las películas de manera eficiente que facilitará la manipulación de datos en el catálogo de películas.

#### 3. Diseño detallado:

Durante esta etapa, se realizó la implementación del diseño propuesto, comenzando por la importación de varios módulos como pandas, el archivo Movie que creamos y ArrayOrderedPositionalList o LinkedOrderedPositionalList que contienen definiciones de clases y funciones necesarias para la simulación del catálogo de películas.

Se define una clase Movie que hereda de la clase ABC (Abstract Base Class) del módulo abc. Esta clase representa una película con los atributos: director, título, año y rating.

La clase Movie implementa un constructor `__init__` para inicializar sus atributos. Utiliza decoradores de propiedades (`@property`) para definir métodos de acceso para obtener los valores de los atributos (director, title, year, rating). También implementa un setter para el atributo rating, que permite validar el valor de la puntuación antes de asignarlo.

El método `__str__` devuelve una representación de cadena de la película, incluyendo su director, año, título y rating. Los métodos de comparación (`__lt__`, `__le__`) permiten comparar películas en función de su director, año y título.

En la clase principal, MovieSimulator, se encuentran funciones esenciales como **load\_movies\_from\_file**, la cual interpreta el archivo de entrada proporcionado para crear instancias de películas. El método **delete\_duplicates** elimina películas duplicadas, conservando solo la versión más reciente de cada película y utilizando únicamente listas posicionales para almacenar y manipular los datos del catálogo.

El mecanismo principal de la simulación se lleva a cabo en el método **execute\_menu**, donde se presenta un menú de opciones al usuario y se ejecuta la acción seleccionada.

La función **metrics** calcula estadísticas sobre las películas, utilizando la biblioteca 'pandas' para realizar estas operaciones, además, creamos una clase Metrics donde se almacenan y acceden atributos como director, título, año y rating.

La función principal, **main**, se encarga de leer desde un archivo de texto que contiene los datos de las películas, cargar estas películas y comenzar la simulación del catálogo de películas.

#### 4. Pruebas y Depuración:

Se realizaron pruebas exhaustivas para cada funcionalidad del programa. Se verificó que la carga de archivos funcionara correctamente y que las películas se almacenaran y ordenaran según lo esperado. Se comprobó que la eliminación de duplicados se realizara de manera adecuada, conservando la versión más reciente de cada película. Se verificó que la visualización de listados y métricas proporcionara resultados precisos y coherentes. Además, se corrigieron errores y se depuró el código para garantizar su correcto funcionamiento en diferentes escenarios.

#### 5. Documentación y Entrega:

Primero redactamos la documentación del código, incluyendo comentarios explicativos y docstrings para cada método y clase. Luego creamos un manual de usuario que describe cómo utilizar el programa y una descripción detallada de las fases de desarrollo realizadas. Finalmente, entregamos el código fuente final junto con la documentación correspondiente.