

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Segundo Parcial		
Apellido ⁽¹⁾ :		Fecha:	3 jul 2025
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Baus / Chavez
División ⁽¹⁾ :	121-1	Nota ⁽²⁾ :	
DNI ⁽¹⁾ :		Firma ⁽²⁾ :	

(1) Campos a completar solo por el estudiante en caso de imprimir este enunciado en papel.

(2) Campos a completar solo por el docente en caso de imprimir este enunciado en papel.

Sistema de Gestión de Catálogo de Películas (V1.1)

Se te pide implementar un sistema de gestión de inventarios para un conjunto de libros. Cada libro tendrá atributos como ID, título, autor y categoría. Además, se debe permitir realizar operaciones como agregar, obtener, eliminar, filtrar, ordenar, guardar y cargar desde archivos.

Requisitos:

1. Clases involucradas:

- **Película:**

Debe contener los siguientes atributos:

- id (entero)
- titulo (String)
- director (String)
- genero (enum Genero)
- La clase Pelicula debe poder ordenarse de manera natural por su id (en orden ascendente).

- **Genero** (enum):

Los posibles géneros de las películas:

- ACCION
- DRAMA
- COMEDIA
- DOCUMENTAL
- TERROR
- ANIMACION

- **Catalogo** (genérico):

Un catálogo genérico que pueda almacenar cualquier tipo de objeto que implemente la

interfaz CSVSerializable.

Debe permitir:

- Agregar, eliminar, y obtener elementos por índice.
- Filtrar según un criterio dado.
- Ordenar los elementos de manera natural o con un Comparator (por ejemplo, por título o director).
- Guardar y cargar desde archivo binario y archivo CSV.

- **CSVSerializable** (interfaz):

Define el método:

- String toCSV()

Las clases que implementen esta interfaz deben poder exportar sus datos en formato CSV.

Además, debe existir un método estático o de clase fromCSV(String) para construir un objeto a partir de una línea CSV.

2. Operaciones del Catálogo:

- Agregar películas al catálogo.
- Obtener películas por índice.
- Eliminar películas por índice.
- Filtrar películas según un criterio (como por ejemplo, por género o por título).
- Ordenar películas:
 - De manera natural (por id).
 - Mediante un Comparator (por ejemplo, por título o director).
- Guardar el catálogo en un archivo binario.
- Cargar el catálogo desde un archivo binario.
- Guardar el catálogo en un archivo CSV.
- Cargar el catálogo desde un archivo CSV.

3. Filtrado:

Implementa al menos dos filtros:

- Filtrar películas por género (por ejemplo, mostrar solo las de género COMEDIA).
- Filtrar películas cuyo título contenga una palabra determinada (por ejemplo, las que contienen "Guerra").

4. Métodos CSV:

- toCSV(): devuelve los datos de la película en formato CSV.
- fromCSV(String csv): crea una película a partir de una línea CSV.

5. Entrega esperada

Debes entregar:

- Código fuente de las clases: Pelicula, Genero, Catalogo<T>, CSVSerializable.
- Métodos para ordenamiento, filtrado, guardado y carga desde archivos.
- Una clase de prueba (main) que verifique el correcto funcionamiento de todas las funcionalidades.
- El proyecto debe reflejar la organización en paquetes vista en la cátedra.
- Las soluciones con código ajeno al visto en la cátedra no serán consideradas válidas.
- Se deben gestionar las rutas desde el paquete config.

```
public static void main(String[] args) {
    try {
        // Crear un catálogo de películas
        Catalogo<Pelicula> catalogoPeliculas = new Catalogo<>();
        catalogoPeliculas.agregar(new Pelicula(1, "El Padrino", "Francis Ford Coppola", Genero.DRAMA));
        catalogoPeliculas.agregar(new Pelicula(2, "La La Land", "Damien Chazelle", Genero.COMEDIA));
        catalogoPeliculas.agregar(new Pelicula(3, "Guerra Mundial Z", "Marc Forster", Genero.TERROR));
        catalogoPeliculas.agregar(new Pelicula(4, "Toy Story", "John Lasseter", Genero.ANIMACION));
        catalogoPeliculas.agregar(new Pelicula(5, "The Social Dilemma", "Jeff Orlowski", Genero.DOCUMENTAL));

        // Mostrar todas las películas en el catálogo
        System.out.println("Catálogo de películas:");
        catalogoPeliculas.paraCadaElemento(/*Acá va una expresión Lambda*/);

        // Filtrar películas por género COMEDIA
        System.out.println("\nPelículas de género COMEDIA:");
        catalogoPeliculas.filtrar(/*Acá va una expresión Lambda*/).forEach(/*Acá va una expresión Lambda*/);

        // Filtrar películas cuyo título contiene "Guerra"
        System.out.println("\nPelículas cuyo título contiene 'Guerra':");
        catalogoPeliculas.filtrar(/*Acá va una expresión Lambda*/).forEach(/*Acá va una expresión Lambda*/);

        // Ordenar películas de manera natural (por id)
        System.out.println("\nPelículas ordenadas de manera natural (por id):");
        catalogoPeliculas.ordenar();
        catalogoPeliculas.paraCadaElemento(/*Acá va una expresión Lambda*/);

        // Ordenar películas por título usando Comparator
        System.out.println("\nPelículas ordenadas por título:");
        catalogoPeliculas.ordenar(/*Acá va una expresión Lambda*/);

        // Guardar el catálogo en archivo binario
        catalogoPeliculas.guardarEnArchivo("src/data/peliculas.dat");

        // Cargar el catálogo desde archivo binario
    }
}
```

```
Catalogo<Pelicula> catalogoCargado = new Catalogo<>();
catalogoCargado.cargarDesdeArchivo("src/data/peliculas.dat");
System.out.println("\nPelículas cargadas desde archivo binario:");
catalogoCargado.paraCadaElemento(/*Acá va una expresión Lambda*/);

// Guardar el catálogo en archivo CSV
catalogoPeliculas.guardarEnCSV("src/data/peliculas.csv");

// Cargar el catálogo desde archivo CSV
catalogoCargado.cargarDesdeCSV("src/data/peliculas.csv", /*Acá va una
expresión Lambda*/);
System.out.println("\nPelículas cargadas desde archivo CSV:");
catalogoCargado.paraCadaElemento(/*Acá va una expresión Lambda*/);

} catch (IOException | ClassNotFoundException e) {
    System.err.println("Error: " + e.getMessage());
}
}
```