

Materia:	Programación II		
Nivel:	2º Cuatrimestre		
Tipo de Examen:	Segundo Parcial		
Apellido ⁽¹⁾ :		Fecha:	4 jul 2025
Nombre/s ⁽¹⁾ :		Docente a cargo ⁽²⁾ :	Baus / Rocha
División ⁽¹⁾ :	122-1	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 Canciones (V1.1)

Se te pide implementar un sistema de gestión de canciones. Cada canción tendrá atributos como ID, título, artista y género musical. Además, el sistema debe permitir agregar, obtener, eliminar, filtrar, ordenar, guardar y cargar desde archivos.

Requisitos:

1. Clases involucradas:

• Cancion

Debe contener los siguientes atributos:

- id (entero)
- titulo (String)
- artista (String)
- genero (enum GeneroMusical)

La clase Cancion debe poder ordenarse de manera natural por su id (en orden descendente).

• GeneroMusical (enum):

Los géneros musicales posibles:

- ROCK
- POP
- JAZZ
- CLASICA
- ELECTRONICA
- REGGAETON

• CatalogoMusical (genérico):

Clase genérica que almacena cualquier tipo de objeto que implemente CSVSerializable. Debe permitir:

- Agregar, eliminar y obtener canciones por índice
 - Filtrar por un criterio
 - Ordenar naturalmente (por id) o con un Comparator (por ejemplo, por título o artista)
 - Guardar y cargar desde archivo binario
 - Guardar y cargar desde archivo CSV
 - **CSVSerializable** (interfaz):
Define el método toCSV().
La clase Cancion debe tener también el método de clase fromCSV(String) para reconstruir desde una línea CSV.
-

2. Operaciones del sistema

- Agregar canciones al catálogo
 - Obtener canciones por índice
 - Eliminar canciones por índice
 - Filtrar canciones por:
 - Género musical (ej: POP)
 - Palabra clave en el título (ej: contiene "love")
 - Ordenar canciones:
 - De manera natural (por id)
 - Mediante Comparator (por ejemplo, por título o artista)
 - 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. Métodos CSV

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

4. Entrega esperada

Debes entregar:

- Proyecto completo.
- Métodos para ordenamiento, filtrado, guardado y carga desde archivos
- Usar el siguiente método (main) que verifique el funcionamiento de todas las funcionalidades. No se puede modificar. Salvo las rutas hardcodeadas y el agregado de lambdas.
- 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 class Main {
    public static void main(String[] args) {
        try {
            CatalogoMusical<Cancion> catalogo = new CatalogoMusical<>();
            catalogo.agregar(new Cancion(1, "Bohemian Rhapsody", "Queen", GeneroMusical.ROCK));
            catalogo.agregar(new Cancion(2, "Billie Jean", "Michael Jackson",
GeneroMusical.POP));
            catalogo.agregar(new Cancion(3, "Shape of You", "Ed Sheeran", GeneroMusical.POP));
            catalogo.agregar(new Cancion(4, "Take Five", "Dave Brubeck", GeneroMusical.JAZZ));
            catalogo.agregar(new Cancion(5, "Canon in D", "Pachelbel", GeneroMusical.CLASICA));

            System.out.println("Catálogo de canciones:");
            catalogo.paraCadaElemento(/*Acá va una expresión Lambda*/);

            System.out.println("\nCanciones de género POP:");
            catalogo.filtrar(/*Acá va una expresión Lambda*/)
                .forEach(/*Acá va una expresión Lambda*/);

            System.out.println("\nCanciones cuyo título contiene 'shape':");
            catalogo.filtrar(/*Acá va una expresión Lambda*/)
                .forEach(/*Acá va una expresión Lambda*/);

            System.out.println("\nCanciones ordenadas por ID:");
            catalogo.ordenar();
            catalogo.paraCadaElemento(/*Acá va una expresión Lambda*/);

            System.out.println("\nCanciones ordenadas por artista:");
            catalogo.ordenar(/*Acá va una expresión Lambda*/);
            catalogo.paraCadaElemento(/*Acá va una expresión Lambda*/);

            catalogo.guardarEnArchivo("src/data/canciones.dat");

            CatalogoMusical<Cancion> cargado = new CatalogoMusical<>();
            cargado.cargarDesdeArchivo("src/data/canciones.dat");
            System.out.println("\nCanciones cargadas desde binario:");
            cargado.paraCadaElemento(/*Acá va una expresión Lambda*/);

            catalogo.guardarEnCSV("src/data/canciones.csv");

            cargado.cargarDesdeCSV("src/data/canciones.csv", /*Acá va una expresión Lambda*/);
            System.out.println("\nCanciones cargadas desde CSV:");
            cargado.paraCadaElemento(/*Acá va una expresión Lambda*/);

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