

PROGRAMACIÓN II: EDA Lineales de Listas.

Tarea Virtual #2

ESTUDIANTE: ANDRES ALFREDO LARREA MANCILLA

CURSO: DSE08

Tema: Colecciones de datos de tipo lista.

Objetivo: Resolución de ejercicios mediante la aplicación de colecciones de datos de tipo la estructura de lista.

Actividad: Implemente un proyecto de consola de POO en Java que permita gestionar una filmoteca con el objetivo de saber la disponibilidad de las películas, además, de otros aspectos de información importantes para la filmoteca. De cada película se desea conocer su título, nombre del actor principal, nombre del director de la película, su género (Comedia, Acción, Ciencia Ficción o Thriller) y año de producción. El sistema debe permitir gestionar la información de todas las películas de la filmoteca y utilizará internamente Colecciones de Datos de Java para almacenar las películas, para ello permitirá añadir películas, eliminarlas y/o ver la lista de todas las películas con que cuenta hasta el momento la filmoteca. El sistema debe permitir los siguientes requerimientos:

- Las operaciones más comunes que haremos en la aplicación van a consistir en
 - añadir una película al final de la lista,
 - eliminar una película dado su identificador (habrá que buscarla en la lista),
 - obtener el listado de todas las películas y recorrerla entera para mostrarlo.
 - *Nota: debe considerar el tipo de colección más apropiada para almacenar esta información.*
- Añadir el código necesario a las operaciones para consultar la lista de películas disponibles. Comprobar que la aplicación funciona correctamente.
- Al obtener la lista de películas almacenadas, mostrarlas ordenadas alfabéticamente. Utilizar para ello los algoritmos que se nos proporcionan en el marco de colecciones.
 - *Sugerencia:* Si internamente estamos almacenando las películas en un tipo de colección sin información de orden, para ordenarlas tendríamos que volcar esta colección a otro tipo de colección que sí que sea ordenable. Es más, aunque internamente se almacenen en una colección con orden, siempre es recomendable volcar los elementos a otra colección antes de devolverla, para evitar que se pueda manipular directamente desde cualquier lugar del código la estructura interna en la que guardamos la información. Además, las películas deberán ser comparables, para que los algoritmos sepan en qué orden se deben ordenar.

Por último, implemente un menú de opciones, en la clase principal del proyecto de consola de Java, para cada uno de los requerimientos descritos anteriormente.

Enlaces (Materiales de Consulta):

- Lectura de las **ideas claves**.
- Ver **Charla Magistral #2**.
- **Lectura del capítulo 6 del libro Mark Allen Weiss:** Estructuras de datos en Java, 4ta edición. **Págs. 225-236, págs. 244-253.** (Sección LO MAS RECOMENDADO).
- **Lectura del capítulo 15 del libro Mark Allen Weiss:** Estructuras de datos en Java, 4ta edición. **Págs. 561-578.** (Sección LO MAS RECOMENDADO).
- **Lectura del capítulo 17 del libro Mark Allen Weiss:** Estructuras de datos en Java, 4ta edición. **Págs. 607-634.** (Sección LO MAS RECOMENDADO).

Orientaciones metodológicas:

- Se valorará la estética, claridad y la correlación con los materiales a consultar.
- Crear un proyecto de consola desde el **IDE Netbeans 8.2** (descargar [aquí](#)) o compilador online de JAVA https://www.onlinegdb.com/online_java_compiler
- Implementar las opciones anteriormente descritas teniendo en cuenta el esquema general de ejecución de algoritmos y el paradigma de programación orientado a objetos.
- Comprobar la solución implementada anteriormente.
- Compactar la carpeta de proyecto con todo su directorio de subcarpetas y archivos del proyecto y, posteriormente, subir el compactado a la plataforma de Eva online.

CODIFICACION DE LA TAREA VIRTUAL

CLASE PELICULA

```
public class Pelicula implements Comparable<Pelicula> {  
  
    private String titulo;  
  
    private String actorPrincipal;  
  
    private String director;  
  
    private String genero;  
  
    private int añoProduccion;  
  
  
  
    public Pelicula(String titulo, String actorPrincipal, String director, String genero, int añoProduccion) {  
  
        this.titulo = titulo;  
  
        this.actorPrincipal = actorPrincipal;  
  
        this.director = director;  
  
        this.genero = genero;  
  
        this.añoProduccion = añoProduccion;  
  
    }  
  
  
    // Getters y setters  
  
    public String getTitulo() {  
  
        return titulo;  
  
    }  
  
}
```

```
}
```

```
// Implementación de compareTo para ordenar alfabéticamente por título
```

```
@Override
```

```
public int compareTo(Pelicula otraPelicula) {
```

```
    return this.titulo.compareTo(otraPelicula.titulo);
```

```
}
```

```
// Método para mostrar la información de la película
```

```
public void mostrarInformacion() {
```

```
    System.out.println("Título: " + titulo);
```

```
    System.out.println("Actor Principal: " + actorPrincipal);
```

```
    System.out.println("Director: " + director);
```

```
    System.out.println("Género: " + genero);
```

```
    System.out.println("Año de Producción: " + añoProduccion);
```

```
    System.out.println("-----");
```

```
}
```

```
}
```

CLASE FILMOTECA

```
import java.util.ArrayList;
```

```
import java.util.Collections;
```

```
import java.util.List;
```

```
public class Filмотeca {
```

```
    private List<Pelicula> peliculas;
```

```
    public Filмотeca() {
```

```
        peliculas = new ArrayList<>();
```

```
}
```

// Método para añadir una película a la filmoteca

```
public void agregarPelicula(Pelicula pelicula) {  
  
    peliculas.add(pelicula);  
  
}
```

// Método para eliminar una película por su título

```
public void eliminarPelicula(String titulo) {  
  
    peliculas.removeIf(pelicula -> pelicula.getTitulo().equalsIgnoreCase(titulo));  
  
}
```

// Método para mostrar todas las películas ordenadas alfabéticamente

```
public void mostrarPeliculasOrdenadas() {  
  
    Collections.sort(peliculas);  
  
    for (Pelicula pelicula : peliculas) {  
  
        pelicula.mostrarInformacion();  
  
    }  
  
}
```

CLASE PRINCIPAL

```
import java.util.Scanner;
```

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Filmoteca filmoteca = new Filmoteca();  
  
        Scanner scanner = new Scanner(System.in);  
  
  
        while (true) {  
  
            System.out.println("==== Menú ====");  
  
            System.out.println("1. Agregar Película");  
  
            System.out.println("2. Eliminar Película");  
  

```

```
System.out.println("3. Mostrar Películas");

System.out.println("4. Salir");

System.out.print("Seleccione una opción: ");

int opcion = scanner.nextInt();

scanner.nextLine(); // Consumir la nueva línea

switch (opcion) {

    case 1:

        System.out.print("Título: ");

        String titulo = scanner.nextLine();

        System.out.print("Actor Principal: ");

        String actor = scanner.nextLine();

        System.out.print("Director: ");

        String director = scanner.nextLine();

        System.out.print("Género: ");

        String genero = scanner.nextLine();

        System.out.print("Año de Producción: ");

        int año = scanner.nextInt();

        scanner.nextLine(); // Consumir la nueva línea

        Pelicula pelicula = new Pelicula(titulo, actor, director, genero, año);

        filmoteca.agregarPelicula(pelicula);

        break;

    case 2:

        System.out.print("Título de la película a eliminar: ");

        String tituloEliminar = scanner.nextLine();

        filmoteca.eliminarPelicula(tituloEliminar);

        break;

    case 3:

        System.out.println("Listado de Películas:");
```

```
        filmoteca.mostrarPelículasOrdenadas();

        break;

    case 4:

        System.out.println("¡Hasta luego!");

        scanner.close();

        System.exit(0);

    default:

        System.out.println("Opción inválida. Intente de nuevo.");

    }

}

}

}
```