

## UNIDAD TEMÁTICA 2: JAVA, PARTE 2

### TRABAJO DE APLICACIÓN 4 – Introducción al proyecto individual

#### Escenario

Se desea realizar una aplicación Java Swing que recupere la lista de películas y los atributos de cada uno desde dos archivos:

1. Peliculas.csv. Contiene la lista de películas, con su ID:

Formato: idPelicula,nombrePelicula,fechaPelicula

Ejemplo:

```
1001,DeadPool,2016
998,50 shades of grey,2015
25,Top Gun,1986
```

2. Actores.csv. Contiene los actores.

Formato: idActor, nombreActor

Ejemplo:

```
253, Ryan Reynolds
200, Tom Cruise
300, Kelly McGillis
450, Dakota Johnson
```

3. PeliculasActores.csv contiene las relaciones entre cada una de las películas y los actores.

Formato: idActor, idPelicula

Ejemplo:

```
1001, 253
998, 450
25, 200
25, 300
```

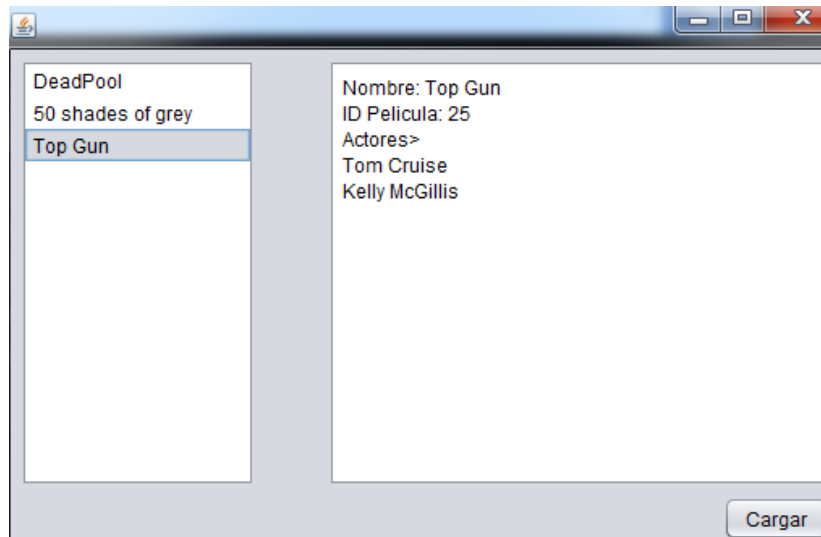
Restricciones y consideraciones:

- Será necesario cargar en **arrays** estos datos, **no** se permiten colecciones de Java.
- Para parsear el archivo **se permite** el uso de la función **Split**. El **separador de campos** será el **carácter coma (',')**

Para la lectura de archivos es necesario utilizar el manejador de archivos genérico provisto por la cátedra. Se encuentra en los recursos de la UT2 o en el siguiente Gist <https://gist.github.com/diegoocampoh/d22f422c1c5f3e1a53c5>

Una vez resuelta la parte de acceso a datos y cargar ambos arreglos, se deberá completar la interfaz gráfica.

La cátedra proporcionará la siguiente clase Java, la cual solo contiene datos de prueba:



1. **Cargar** – Ya se encuentra implementado, al hacer clic carga la lista de películas mostrándola a la izquierda.
2. **Lista de películas.** Se provee un método *que al seleccionar una película carga toda la información en el bloque de texto de la derecha.*

## Dinámica del trabajo

### Paso 1:

Este paso lo realiza **todo el equipo junto**.

En esta instancia deberán definir la interfaz o contrato de la lógica principal a utilizar para separar claramente las **responsabilidades** y **desacoplar** el código. Para más información lean este artículo: [http://en.wikipedia.org/wiki/Loose\\_coupling](http://en.wikipedia.org/wiki/Loose_coupling)

Las dos principales responsabilidades son:

1. **Lectura de archivos y carga de estructura de datos**
2. **Interfaz gráfica**

Es necesario definir una interfaz antes de empezar a trabajar. Esta interfaz será la del **adaptador** de datos de películas, cuyo **contrato** debería contener las siguientes **responsabilidades**:

- **Obtener películas**
- **Obtener actores**
- **Obtener peliculasActores**

Al implementar esa interfaz para leer los datos, **se exponen tres métodos**:

- `Void cargarDatos();`
- `String[][] obtenerPeliculas();`

- `String[][] obtenerActores();`
- `String[][] obtenerPelículasActores();`

De esta forma, la **GUI se abstrae** de cómo se hace la carga de datos, simplemente debe inicializar el objeto que cumple con el contrato, llamar a **cargarDatos** y luego leer los datos. La GUI no debería enterarse si los datos se traen de un csv, base de datos o webservice.

Se sugiere e incluye la siguiente interfaz:

```
public interface IAEDMovieAdapter {
    void cargarDatos();
    String[][] obtenerPelículas();

    String[][] obtenerActores();

    String[][] obtenerPelículasActores();
}
```

Se sugieren **matrices** para poder vincular **id** de **película** con el **nombre** del actor e **id** de actor con **la lista de películas actores** ordenada por fecha. Si desean implementar un tipo de datos **Actor** o **Película**, por ejemplo, es válido, siempre y cuando **no usen colecciones**. En tal caso pueden alterar el contrato.

De esta forma desde la GUI se hará algo similar a lo siguiente:

```
IAEDMovieAdapter adaptor = new AEDMovieAdapter();
adaptor.cargarDatos();
adaptor. ....
```

Todo lo que consideren que debe ser parte del contrato se agrega en el mismo; detalles de implementación como métodos privados **no deberían ser parte de éste**.

### ***Paso 2:***

El Equipo se dividirá en 2 subequipos para realizar este trabajo.

#### ***Subequipo A – Diseñadores de UI***

Este sub-equipo deberá implementar los métodos que están comentados con **//TODO:** , sin tener que preocuparse por la lógica inherente a la lectura del archivo .

#### ***Subequipo B – Desarrolladores de lógica principal***

Este sub-equipo deberá realizar toda la lógica totalmente desacoplada de la UI, para que pueda ser usada tanto desde una aplicación Swing, Web, mobile, etc.

**IMPORTANTE: No se admiten colecciones, solo se pueden utilizar arrays.**

### ***Paso 3:***

El Equipo trabajando todos juntos:

- Integrar y sincronizar todo el código.
- Ejecutar y probar el programa con distintos conjuntos de datos (se proveen varios conjuntos, de diferentes extensiones)

**TODOS LOS FUENTES Y CASOS DE PRUEBA DEBERÁN ESTAR SINCRONIZADOS EN SUS REPOSITORIOS ANTES DE LA HORA 21:20**