

IMBD21

Informe del proyecto

David García Curbelo

El presente proyecto tiene como objetivo facilitar al usuario la lectura, escritura y almacenamiento de los datos, en este caso concreto procedentes de películas, con sus respectivos actores, directores, géneros, etc.

Presentando una arquitectura MVC, han sido añadidos al paquete data los módulos habituales, y los necesarios para el correcto funcionamiento de la aplicación: Filmoteca, Pelicula, Actor, Director y por supuesto el modelo. Además, en el paquete view me he servido de un ejemplar auxiliar al archivo View, con la intención de que éste sólo controle la interfaz de los menús principales y el programa de arranque e importación de archivos. El visor auxiliar (ViewAux) se encarga de controlar las funciones de cada uno de los submenús que se encuentran en la clase View. De este modo la revisión de código queda más organizada y de más fácil acceso.

Se han aportado a nivel general un alto contenido de mensajes de error, para que el usuario pueda identificar no sólo los errores sino también su ubicación para la posterior corrección. Así mismo, y con idéntica finalidad, se han añadido mensajes de información al inicio del programa, para informar acerca de los archivos que han sido importados y el estado de éstos.

Por ello he añadido, en un comentario, una parada temporal en el programa inicial (runMenu) para que el usuario pueda ver los mensajes de información referentes a la importación de archivos que se acaba de realizar. Dicha parada no es estrictamente necesaria (además, no está en funcionamiento. Se encuentra comentada en la línea 26 de View.java), y se puede eliminar en el futuro junto

con el `clear()` que se ejecuta cada vez que se recorre el bucle `while` del `runMenu` en la clase `View`. Dicha parada ha requerido un `throws InterruptedException` en el `runMenu` (también en un comentario en la línea 8) para informar de que se dará lugar una parada temporal de 5000 milisegundos, que ha sido ejecutada mediante `Thread.sleep(time)` de la línea 26. Recaltar que esta parada no está en funcionamiento, ya que se encuentra fuera del temario del programa.

Se ha procurado un uso máximo de `biblioteca.jar`, mediante la cual ha sido usado el método `printToScreen3()` en la clase `ViewAux` para la impresión de listas requeridas para el proyecto, así como los métodos de importación y exportación de archivos.

Dicha importación ha sido estructurada de tal forma que se dé prioridad al archivo de tipo binario, y en caso de que éste no exista, o que se pueda encontrar dañado, se intentará la importación del archivo de texto (para priorizar el funcionamiento del programa antes que la importación de archivos binarios). En caso de que no se pueda encontrar ninguno de los archivos, se finalizará la ejecución del programa y se mostrará un mensaje por cada importación realizada, y se indicará qué archivo se ha importado de cada tipo, y cuál de ellos no se ha podido importar (con su respectivo motivo).

Una vez importados los archivos, se procede a la creación de un ejemplar de filmoteca, que se encargará de almacenar todos los datos de las películas, actores y directores. Esta estructura de datos se ha proporcionado con el fin de que en un futuro el programa pueda gestionar varias filmotecas a la vez, simplemente añadiendo más ejemplares de filmoteca. Así, por ejemplo, podremos tratar con una filmoteca de películas de cine, y otra de cortometrajes, pudiendo acceder a cada una de ellas por separado.

Notar que cualquier modificación que se realice en los ejemplares propios de películas, actores o directores, (ya sea añadir, borrar o modificar) se verán reflejados en la filmoteca, y posteriormente en el archivo binario exportado, que guardará el estado de la filmoteca en el último arranque que tuvo el programa.