Documentación de las clases:

Clase **PanelPelicula**:

Esta clase tiene como propósito interpretar el formato de las líneas del archivo y generar un pequeño panel con los datos correspondientes. Para implementar esto, se crean los siguientes métodos y eventos:

- `PanelPelicula(String info)` Éste es el único constructor que tiene la clase. El string que se le entrega contiene la información el nombre de la película, la duración y la ruta de acceso.
- `String info()` Este método retorna un string ya procesado, que contiene toda la información del panel, para que pueda ser guardado en un archivo
- `ActionListener AlBut` Este evento es invocado cuando se presiona el botón para buscar la ruta del video. Para realizar esto, se usa un objeto JFileChooser, el cual despliega un panel para poder elegir el archivo deseado.

Clase **videoAdmin**:

Esta clase es el programa en sí, por lo que su estructura es un poco más compleja. Sus métodos y eventos son los siguientes:

- `public static void main(String[] args)`El método main tiene como objetivo iniciar el programa. Aquí lo que hace es chequear los parámetros y llamar al constructor de videoAdmin.
- `public videoAdmin(String name,String arch)` Éste es el constructor que tiene la clase. Toma como primer parámetro el título de de la ventana, y como segundo, el nombre del archivo de dónde sacar los videos. Después crea la ventana, le agrega los botones para deshacer, actualizar y agregar, y agrega el panel que va a contener a los videos. Después de eso, llama al método `LeeArchivo()`
- `public int LeeArchivo()` Se preocupa de leer la información del archivo y de ir creando los paneles `PanelPelicula`, los cuales van conteniendo la información acerca de los videos. Además, los agrega al scrollpanel, junto con un botón para eliminarlos.
- `public void EscribeArchivo()` Se preocupa de recorrer todos los `PanelPelicula` para pedir la información necesaria, usando el método `info()`. Con la información obtenida aquí, se va regenerando el archivo de datos.
- `ActionListener AActualizar` Controla el botón actualizar. Cuando este botón es presionado, se llama al método `EscribeArchivo()`.

- `ActionListener ADeshacer` Controla el botón Deshacer. Cuando este botón es presionado, se llama al método `LeeArchivo()` para restablecer la lista de videos.
- `ActionListener ANuevo` Controla el botón Nuevo. Cuando este botón es presionado, se inserta un nuevo `PanelPelicula` vacío, al final de la lista.
- `ActionListener ABorrar` Controla todos los botones de borrar video. Cuando alguno de los botones es presionado, se identifica cuál fue, y se borra el botón y el `PanelPelicula` asociado, y aprovecha de mover todo el resto de los elementos que se encuentran a continuación, un espacio hacia abajo.
- `class MyActionListener` Esta clase implementa un ActionListener que se preocupa de el cerrado de la aplicación cuando se presiona el botón de cerrado.