

# Ejercicios de depuración y logging

## Índice

1 Depuración de código (1.5 puntos).....	2
2 Logs al leer ficheros (1.5 puntos).....	2

## 1. Depuración de código (1.5 puntos)

Vamos a depurar la función para el cálculo del importe total de una venta, que se encuentra en el proyecto `lja-debug` de las plantillas de la sesión. La función `calculaImporteTotal` (de la clase `Venta`) toma como entrada los datos de la venta que comprenden una serie de líneas de venta, con el precio unitario de cada producto, el tipo del producto, y el número de unidades vendidas, y devuelve un valor numérico con el importe total de la misma. Según el tipo de producto que tengamos se le aplicará un IVA distinto en el precio. Se pide:

- Añadir un breakpoint al comienzo de la función. Ejecutar paso a paso y monitorizar el valor de la variable `total` y de las expresiones `lv.getCantidad() * lv.getPrecio()` (precio antes de aplicar el IVA) y `lv.getCantidad() * lv.getPrecio() * (lv.getTipo().getIva())` (precio con el IVA incluido). Comprobar el valor que toman en cada iteración y detectar posibles errores en el código. Hacer las correcciones oportunas.
- Añadir un breakpoint condicional que sólo se ejecute cuando la cantidad del producto actual sea mayor que uno.
- Probar a cambiar en tiempo de ejecución la cantidad de uno de los productos.

## 2. Logs al leer ficheros (1.5 puntos)

Vamos a añadir logs a nuestra aplicación de filmoteca. Se pide:

- Añadir mensajes de depuración (DEBUG) en `FilePelículaDAO` que indiquen qué va haciendo el programa en cada paso, y mensajes de error (ERROR) cuando ocurra un error (en los casos en los que se lance una excepción).
- Modificar la configuración del logging, mediante un fichero de propiedades, para que:
  - Que todos los mensajes vayan a la consola, con el formato simple.
  - Además, que la salida de mensajes de tipo ERROR se vuelque a un fichero `errores.log`, con el siguiente formato:

```
[día/mes/año hora:minuto:segundo] - tipo error - mensaje
```

Utilizaremos la librería *commons logging* para gestionar los *logs*, y dentro de ella indicaremos que se utilice *log4j*.

### Nota

Para crear los ficheros de configuración necesarios podemos basarnos en los ejemplos que se encuentran en el proyecto `lja-debug` de las plantillas de la sesión (utilizar los ficheros de configuración por defecto que tenemos en el directorio `resources`). También podemos coger de este proyecto las librerías de *logs* necesarias.

- Probad el programa creando, eliminando y consultando la lista de películas. Forzar a que se produzcan errores para que se muestren los *logs* de error.
- Mediante el fichero de configuración, haced que el fichero *errores.log* no pueda exceder de 1 KB (deberéis utilizar un *RollingFileAppender* en lugar de un *FileAppender*) y aseguraos de que no se sobrescriba su contenido tras cada nueva ejecución. Probad a ejecutar el programa varias veces, provocando errores que se guarden en dicho fichero, y comprobad cómo se rotan los mensajes.
- Probad a que se guarden 2 copias de respaldo del fichero *errores.log*, y generad mensajes suficientes en él para que sobrepase el tamaño de un fichero. Observad cómo se generan los ficheros de respaldo, y qué nombres reciben.

