

Enunciado

Se quiere construir un programa que permita manejar la información de un conjunto de películas. De una película se conoce el nombre, el año de producción y un resumen de ella. La aplicación debe permitir cargar un gran volumen de películas que se encuentran almacenadas en un archivo de texto (cupi2.uniandes.edu.co/cursos/datos/201210/nivel14/plot.list.gz) y permitir realizar una serie de operaciones sobre ellas. La aplicación debe permitir crear índices de películas, a partir de palabras de su resumen/descripción. Un índice es una estructura de datos que permite clasificar objetos a partir de un dato/información que es común a un subconjunto de ellos. En la aplicación de películas, se quiere crear índices sobre palabras completas que aparezcan en el resumen/descripción de las películas.

La información de la aplicación debe ser persistente y el proceso debe ser completamente transparente para el usuario. En caso de cualquier error en la ejecución de la aplicación, ésta debe presentar un diálogo con un mensaje claro que explique la razón del problema.

El problema

El objetivo de este ejercicio es analizar, diseñar e implementar un sistema de información de películas, utilizando unas estructuras de datos que tengan en cuenta las restricciones que se definen más adelante. Asegúrese de leer completamente el enunciado antes de resolver el ejercicio.

Requerimientos funcionales

El programa debe soportar los siguientes requerimientos funcionales:

- R1.** Cargar la información de las películas desde el archivo de texto dado.
- R2.** Buscar películas por su nombre o parte de él y mostrar toda su información (Se debe poder navegar por el resultado).
- R3.** Buscar películas por su descripción o parte de ella y mostrar toda su información (Se debe poder navegar por el resultado).
- R4.** Buscar las películas de un año dado y mostrar toda su información (Se debe poder navegar por el resultado).
- R5.** Buscar las películas que pertenecen a un rango de años dado y mostrar toda su información (Se debe poder navegar por el resultado).

R6. Crear un índice de películas e indexar las películas pertenecientes a este índice. La palabra de clasificación de un índice No debe ser sensible a mayúsculas ni minúsculas.

R7. Ver todos los índices existentes ordenados por orden alfabético incluyendo el número de películas que tiene cada uno.

R8. Ver todas las películas que pertenecen a un índice dado y mostrar toda su información (Se debe poder navegar por el resultado).

R9. Mostrar la lista de los 10 índices más populares (los que más películas tienen asociados) en orden descendente por número de películas. Cada índice debe mostrar su número de películas..

Restricciones de diseño

1. El diseño se debe hacer **MINIMIZANDO el tiempo de ejecución de todas las operaciones y el espacio que ocupan las estructuras de datos, garantizando que el programa pueda evolucionar.**
2. Debe usar **TABLAS DE HASHING** al menos para un requerimiento con una función de hashing definida por usted.
3. Las estructuras de datos deben ser genéricas y sus interfaces deben extender **Iterable**.
4. Debe crear su propia librería de estructuras de datos. Para ello es necesario que implemente en un proyecto separado todas las estructuras que va a utilizar. **Se debe utilizar ant para la compilación y empaquetamiento de la librería.**
5. Se deben incluir las **clases de prueba** JUnit que correspondan a los escenarios y los casos de prueba para todas las estructuras de datos que utilice.
6. El diseño de la solución debe estar desacoplado con interfaces.
7. El sistema debe utilizar ant para su compilación y **ejecución**.
8. En particular, se deben justificar las decisiones de diseño tomadas, se debe explicar claramente en un **documento de diseño** por que se escogieron las estructuras de datos utilizadas, teniendo en cuenta los requerimientos descritos anteriormente.
9. El sistema debe ser **persistente**. Recuerde que el objetivo del ejercicio es resolver los requerimientos con las estructuras en memoria principal, por este motivo no se acepta el uso de bases de datos. Puede utilizar archivos de texto o serialización para persistir el modelo del mundo.
10. La interfaz gráfica debe ser implementada utilizando componentes de awt/swing.
11. No se puede utilizar Java Collections ni Cupi2 Collections.

Entregables

Semana 1 (10%): Se debe traer para el laboratorio 1 el documento de Análisis y Diseño completo y un prototipo de la Interfaz Gráfica.

Semana 2 (10%): Se debe traer para el laboratorio 2 la implementación de las estructuras de datos a utilizar, en un proyecto aparte, con sus respectivas pruebas unitarias y empaquetas mediante el uso de ant. La implementación de la Interfaz Gráfica y funcionando. Adicionalmente, la aplicación con el requerimiento R1 y algún otro requerimiento funcional.

Semana 3 (80%): Entrega final completa – documento de Análisis y Diseño, implementación de Estructuras de Datos y uso de ant para empaquetamiento, implementación de Interfaz Gráfica, implementación de Aplicación con requerimientos funcionando y uso de ant para compilación/ejecución.

El mejor ejercicio de cada sección tendrá un bono especial sobre la nota del nivel.