EDA - ETSInf - Curso 2018-2019

Práctica 3 - Parte 1: Uso de un Map de términos de una Biblioteca Virtual

Departamento de Sistemas Informaticos y Computación
Universitat Politècnica de València

1. Objetivos formativos y trabajo previo

El objetivo final de esta práctica es el uso de una Tabla de Dispersión o *Tabla Hash* para implementar el *Map* de términos (palabras) que representa un índice en una colección de libros digitalizados en formato de texto. En esta primera parte el alumno utilizará un `Map` para representar el índice de términos de la colección de libros. En concreto, debe identificar clave y valor e implementar las operaciones para construir el `Map`, a partir del análisis de los textos y para consultar la información relacionada con un términos dado.

Además se reforzarán los objetivos transversales a todas las prácticas de la asignatura y que están relacionados con la calidad de los programas desarrollados: utilización de paquetes para facilitar la organización, reutilización y mantenimiento del software, utilización de los mecanismos de herencia y genericidad que proporciona el lenguaje, elaboración de juegos de prueba para validar código y generación de documentación asociada al código desarrollado. Para aprovechar al máximo la sesión de laboratorio, antes se debe realizar una lectura comprensiva de este boletín y del código de las clases que se proporcionan a través de PoliformaT.

2. Índices de palabras para un conjunto de libros

En la actualidad existe una enorme cantidad de información disponible a través de documentos en soporte digital. El problema de tener tanta información es que no se dispone de tiempo suficiente para consultarla de manera exhaustiva. Por ello, es imprescindible idear métodos eficaces para centrar la búsqueda en aquellos documentos que puedan contener la información que nos interesa.

Un ejemplo muy claro de ello son los buscadores de Internet tipo Google. Si se dispone de un repositorio de documentos de texto, en nuestro caso una biblioteca de libros, nos puede interesar un sistema para buscar términos en ella, de manera que nos guíe a las lineas de los documentos y libros que los contengan. En la figura 1 vemos algunos ejemplos de la aparición de la palabra *datos* en dos de los libros que se tienen disponibles: *Acceso Abierto* y *Aprendiendo Java y Programación Orientada a Objetos*: ha aprecido en las líneas 305, 435, 440, 452, 538, etc. en el primer libro y en las líneas 26, 29, 33, 44, etc. en el segundo.

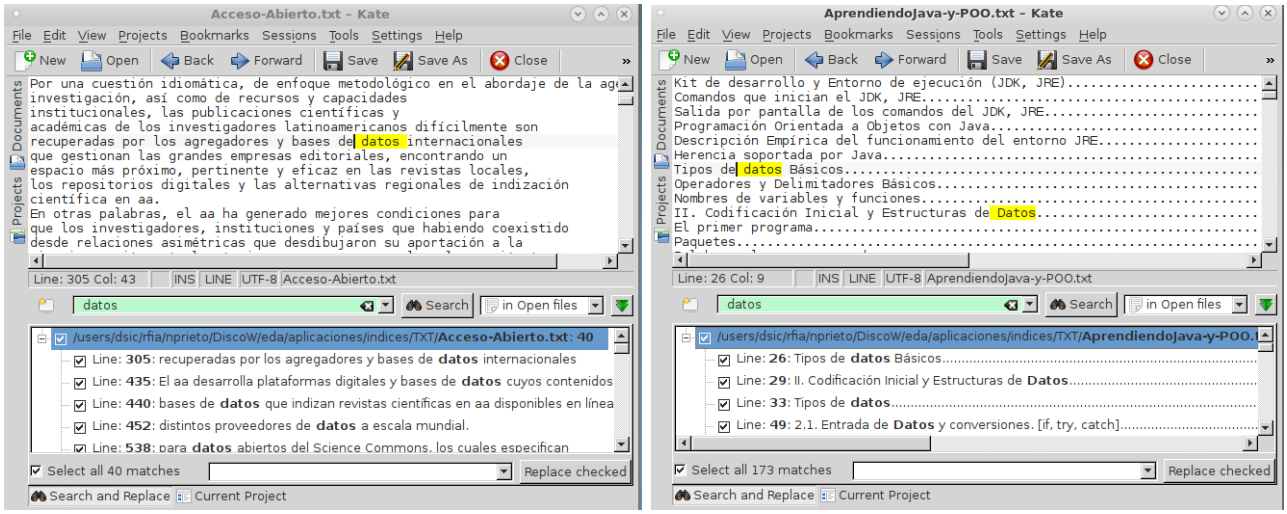

Figura 1: Ejemplos de localización de la palabra datos en dos documentos

El sistema que se va a diseñar en esta práctica utilizará un *Map*, implementado mediante una *Tabla Hash*, en el que las claves serán los términos (o palabras) y los valores los nombres de los libros y líneas en las que éstos aparecen. Las claves se representan utilizando la clase predefinida `String` y como `hashCode` el que define esta misma clase (ver `https://docs.oracle.com/javase/8/docs/api/java/lang/String.html#hashCode--`). Además, la aplicación tendrá las siguientes clases:

- `Indice`: representa la aparición de una palabra; tiene dos atributos, un `String` que contendrá el nombre del libro donde aparece, y un entero que contendrá la línea donde se encuentra la palabra en dicho libro.
- `Indexador`: tiene como atributo el `Map<String, ListaConPI<Indice>>`, en el que las claves son las palabras y los valores la información asociada a cada una de ellas, en este caso, una `ListaConPI<Indice>` que contiene los índices para dicha palabra, es decir, los títulos de los libros y las líneas donde esta palabra aparece.
- `GUIBiblioteca`: aplicación gráfica simple que permite cargar la colección de libros (indexarla) y buscar una palabra en dicha colección. En la figura 2 se puede ver un ejemplo de su ejecución, a la izquierda la interfaz gráfica y a la derecha la ventana de ejecución con información adicional.

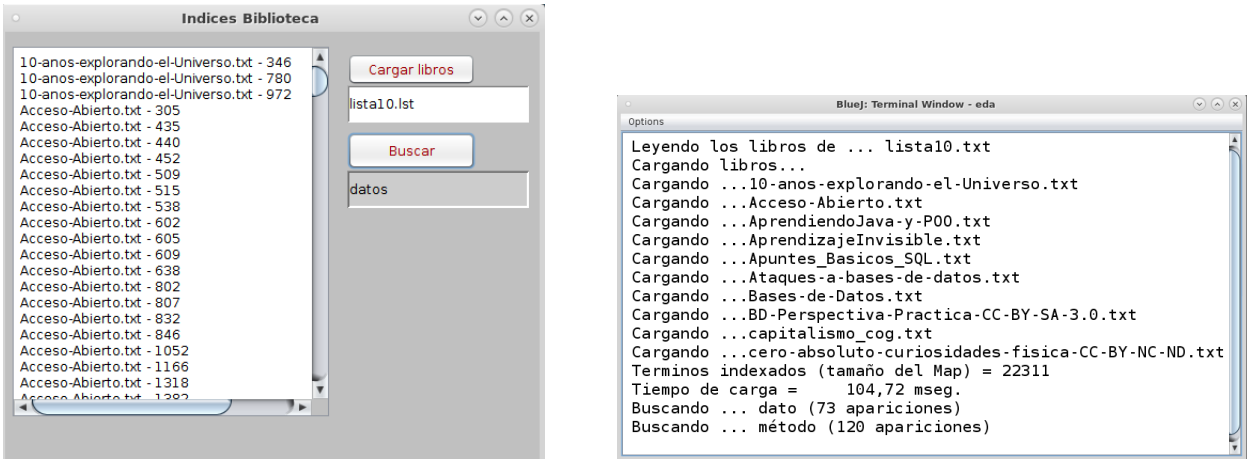

Figura 2: La interfaz gráfica `GUIBiblioteca` y terminal de ejecución

Actividad #1: Organización de paquetes y clases

Utilizando el material disponible en *PoliformaT*, realiza las siguientes acciones en el proyecto BlueJ *eda*:

1. Agrega al paquete *librerias.estructurasDeDatos.modelos* la interfaz `Map` y compílala. Recuérdese que *Map* es un modelo orientado a la búsqueda por clave en una colección de datos.

2. Crea el nuevo paquete *estructurasDeDatos.deDispersion* en *librerias*.

3. Copia los ficheros `TablaHash.class` y `EntradaHash.class` en la carpeta `deDispersion`.

4. Crea un nuevo paquete, `aplicaciones.biblioteca`, que contendrá la aplicación destinada a la obtención, para una palabra y una colección de libros, de las líneas de los libros en las que aparece. En él añade las clases `Indice`, `Indexador` y `GUIBiblioteca`.

Actividad #2: la clase `Indexador`

Define el atributo Map de la clase `Indexador` e inicialízalo en su constructor. Además, completa los siguientes métodos de la clase:

- `cargarLibro`: para cada palabra del libro debe actualizar el índice asociado, si la palabra aún no ha aparecido debe crear la lista de índices vacía antes de insertar.

- `indiceDe`: debe recuperar del `Map` la lista de `Indice` para la palabra dada y construir una `ListaConPI` de String que será el resultado del método.

Actividad #3: uso de la aplicación `GUIBiblioteca`

Comprueba el correcto funcionamiento de la aplicación `GUIBiblioteca`. Para ello debes:

1. Copiar los ficheros `lista.txt` y `lista10.txt` en la carpeta `biblioteca`. Estos ficheros contienen, respectivamente, los nombres de todos los documentos disponibles en la biblioteca o sólo `10` de ellos. Los documentos se encuentran en la carpeta común `asigDSIC/ETSINF/eda/libros/TXT`.

2. Ejecutar el `main(String[] args)` de la clase `GUIBiblioteca`, utilizando los dos listados de libros; en el primero se indexan `105986` términos mientras que en el segundo sólo `22311`.

3. El fichero `TablaHash.class` que se te ha proporcionado no realiza `rehashing`. Para ver el efecto que tiene la capacidad del array, soporte de la *Tabla Hash*, modifica la constante `TALLA_VOCABULARIO` de la clase `Indexador` para darle un valor mucho menor (por ejemplo 100) y comprueba como el tiempo de construcción del *Map* se incrementa de forma importante. Una vez visto, deja el valor original para la constante `TALLA_VOCABULARIO`.