**Análisis del funcionamiento de Boocle**

Se ingresa una frase/palabra a buscar…

SearchServlet gestiona todas las tareas.

Se carga el objeto Dictionary desde un archivo.

Analiza la similaridad de éste

Básicamente el objeto Diccionary tiene un vector con el path de todos los archivos, un hashtable con todas las palabras, un hash con las frecuencias inversas y un hash con los archivos por palabra.

Posteriormente levanta desde un archivo un objeto DocumentIndex que es un hastable, la key son los términos y los datos son un hashset que son los path de los documentos en los que aparece.

**Arquitectura del buscador.**

El buscador se comprenderá por dos módulos. Uno será el indizador que estará corriendo en el servidor constantemente calculando los pesos de las palabras en los documentos y las frecuencias inversas.

Por otro lado tendremos el buscador que recuperará esta información generada por el indizador, evaluará la similaridad y devolverá una lista ordenada con los documentos resultantes de la búsqueda.

**Conceptos básicos del indizador:**

Vocabulario: Se implementará con un hashtable

Lista de posteo: Se debería implementar en un archivo

Se le configura un directorio y debe recorrer recurrentemente todos los archivos.

Por cada archivo:

Leer palabra por palabra

Por cada palabra

Llevar a minúscula, sacar puntuaciones

Verificar si no es stop Word

Actualizar hash de vocabulario

Crear / actualizar lista de posteo para ese término

Fin por cada palabra

Fin por cada archivo

Ordenar cada lista de posteo

El archivo de la lista de posteo va a ser un archivo que se comprenderá de objetos NodoListaPosteo, los cuales tendrán la siguiente estructura: (frecuencia int ,archivo int, next long). La frecuencia indicará la cantidad de veces que aparece el termino en el documento, el archivo será un entero que apunte a la posición del nombre del archivo, el cual va a estar dentro del archivo de documentos y el next va a ser un indicador al próximo nodo de posteo dentro del archivo para dicho termino, para cumplir el objetivo de utilizar dicho documento como una lista enlazada, el next del último nodo de posteo va a ser -1, lo cual nos indicará que es el fin de la lista enlazada para dicho termino.

La generación de esta listas de posteo será a medida que se vayan leyendo los términos por cada archivo, por lo que las listas estarán desordenadas cuando se termine de leer el último archivo.

Posteriormente se deberá recorrer por cada término cada lista de posteo, levantándola a memoria, ordenándola y grabándola en otro archivo ordenada.

Al finalizar esto ya tenemos todo listo para empezar a buscar.

Archivos que genera

Docs.dat

Vocabulario.dat

ListaPosteo.dat

PosteoOrdenado.dat

Hacer la lógica de cómo guarda y como recupera las listas de posteo!

**LOGICA DE BUSQUEDA**

1. Genero un objeto búsqueda
2. Seteo el criterio al objeto Busqueda (String Buffer)
3. Seteo la cantidad de resultados a obtener
4. Materializo el Vocabulario (esto se podría hacer al arrancar el buscador y mantenerlo en memoria)
5. Creo un GestorBusqueda y llamo a buscar