Informe del Proyecto

Estudiante: Carlos Mario Chang Jardínez C121

Para realizar este proyecto he utilizado arrays, listas y diccionarios según los he visto necesario para mejorar la efectividad de la búsqueda. Lo primero que hice fue hacer una clase Build para hacer unas cuantas operaciones con la base de datos que luego harán que el usuario tenga una mejor experiencia. En la clase Build tengo implementado los métodos: HacerTF\_BuscarSnipet el cual devuelve 2 diccionarios, uno con cada una de las palabra y su TF y el otro con cada palabra y un arreglo de snipets del tamaño de la cantidad de documentos donde en cada posición tendrá un snipet de 300 caracteres que la contiene. También cargo un arreglo de floats que contiene el idf de cada palabra. Y por último cargo un diccionario que contiene muchas palabras y sus respectivos sinónimo. Todo eso se ejecuta antes de levantar la página. Una vez cargada la página ya tendríamos hecha la mayoría de las cosas. El método Query de la clase Moogle tiene que realizar sencillas operaciones. Una vez que le es enviado el query, se ejecuta un método que se llama: PicarString, que va a devolverme un arreglo con las palabras del string que vamos a buscar, una matriz que contiene la información de los operadores introducidos en el query y la sugerencia que calculé utilizando la distancia de Levenstein. Después se ejecuta el método HacerTFIDF\_Sistema que devolverá un arreglo de floats con la puntuación de cada documento utilizando el algoritmo TF\*IDF. Más tarde el método: Cargar snipet se encarga de almacenar en un arreglo de strings los snipets de cada documento en los que se encontraron resultados. El siguiente método es CargarTitulos, cuyo nombre ya es bastante sugerente. Ahora que tenemos tres arreglos, uno con el TFIDF, uno con los snipets y uno con los títulos, se ejecuta el último método: DevolverBusqueda que se encargará de almacenar cada elemento en un objeto SearchItem.