Proyecto de Programación I. Moogle!

Nombre: Ana Paula González Muñoz

Grupo: 121

El proyecto Moogle! consiste en un sistema de recuperación de información que basa su funcionamiento central en el algoritmo TF-IDF y en la implementación de modelos vectoriales para garantizar búsquedas más eficientes y acertadas.

Se emplearon un conjunto de lenguajes de programación para lograr la mejor vinculación entre los elementos que conforman el código. Como principal motor en la creación y el trabajo con los algoritmos, se empleó el lenguaje C#, mientras que la interfaz gráfica se encuentra codificada en una plataforma web.

Para garantizar una edición sencilla y la legibilidad del código desarrollado, se utilizaron en el proyecto las clases:

- ✓ *preSearch*, que se encarga de hacer todos los cálculos pertinentes antes de realizar la búsqueda. Encontramos los métodos:
  - *ObtenerRuta*: se encarga de obtener la ruta de cada txt que se encuentran en la base de datos.
  - *Normalizar*: se encarga de normalizar los textos (eliminar tildes, espacios, caracteres que puedan hacer ruido).
  - *GuardarTxt*: se encarga de guardar los txt normalizados de la base de datos para poder trabajar con ellos.
  - *TfIdf*: se encarga de calcular Frecuencia Inversa – Frecuencia Inversa del Documento (TF*IDF) para darle un valor a las palabras según su importancia en la base de datos y en los documentos.

- ✓ *search,* se encarga de hacer todos los cálculos pertinentes durante la búsqueda. Encontramos los métodos:
  - *obtener query:* se encarga de cargar y normalizar (sin eliminar los operadores) la búsqueda que introduce el usuario.
  - *guardarOperadores:* se encarga de guardar las palabras con operadores (~, ^, *, !) y eliminar los operadores de la búsqueda.
  - *Tfquery:* se encargad de calcular la Frecuencia Inversa (TF) de las palabras de la búsqueda que introduce el usuario.
  - *querySinRepetir:* se encarga de eliminar las palabras repetidas de la búsqueda que introduce el usuario
  - *queryTfIdf:* se encarga de multiplicar la Frecuencia Inversa (TF) de cada palabra la búsqueda por la Frecuencia Inversa del Documento (IDF) general de la misma palabra.
  - *OperadorDeImportancia:* se encarga de modificar el TF*IDF de las palabras de la búsqueda que contengan dicho operador.

- *magnitudDeLaQuery:* se encarga de calcular la magnitud $\sqrt{A^2+B^2+\ldots+N^2}$ (donde A,B…N es el TF*IDF de la búsqueda).
- *diccionarioDeSimilitudDeCoseno*: se encarga de convertir los documentos y la búsqueda del usuario en vectores del plano para saber cuales documentos son mas relevantes. Se divide por partes:
    - *o* vector documento: crea el vector documento multiplicando el TF*IDF de las palabras del documento.
    - *o* suma punto: sumatoria de los TF*IDF de la búsqueda y del documento.
    - *o* magnitud del documento: se encarga de calcular la magnitud $\sqrt{A^2+B^2+\ldots+N^2}$ (donde A,B…N es el TF*IDF del documento).

- *similitudDeCosenoOrdenado*: en una lista se guardan los documentos que más coincidencias tienen con la búsqueda del usuario.
- *Operadores*: se encarga de realizar las operaciones pertinentes según los operadores que tenga la búsqueda:
    - o ! no debe aparecer la palabra: se eliminan los documentos que contengan esa palabra del diccionario que almacena todas las similitudes de coseno.
    - o ^ tiene que aparecer: se eliminan los documentos que no contengan esa palabra en el diccionario que almacena todas las similitudes de coseno.
    - o ~ cercanía: se aumenta la similitud de coseno mientras más cercanas sean las palabras.
- *OrdenarQuerySegunTfIdf*: se encarga de ordenar las palabras de la query según su importancia. Es un método auxiliar para los snippets.
- *Snippets*: se encarga de mostrar una parte del documento donde se encuentra la palabra con mayor coincidencia de la búsqueda en el mismo.