Proyecto Moogle!

Victor Manuel Castillo Tamayo C111

Julio, 2023

Abstract

`Que es Moogle!?`

Esta aplicación web Moogle!, desarrollada con .NET core 6.0, en el lenguaje CSharp, tiene la funcionalidad de buscar una palabra o frase (query) insertada por el usuario, en cierto grupo de documentos .txt, y mostrar el resultado en su interfaz. Si el resultado, no es encontrado, no se mostrara nada en la interfaz. Se le sugeriran palabras en relacion a su busqueda, que posiblemente tendran mejores resultados. La búsqueda devolverá los documentos donde se encuentre el "query" insertado por el usuario y una porción de este, mostrando donde fue encontrado. El usuario puede editar o sustituir los documentos contenidos en la carpeta "Content", siempre y cuando se respete la condición de estar en formato .txt .

1 Introduccion

A modo de presentacion este documento esta dirigido a la explicacion del funcionamiento del codigo tras el programa, resumiendo el metodo utilizado para la busqueda del query y la obtencion de la porcion del documento donde se encuentra este. Para esto damos a conocer las distintas clases y metodos utilizados para el correcto funcionamiento de este.

2 Como correr el programa?

Al abrir la terminal en la carpeta del proyecto

Sistema Operativo	correr en la termnal
Linux	make dev
Windows	dotnet watch run –project MoogleServer

3 Desarrollo

Este proyecto esta implementado de la siguiente forma:

- Carga de archivos: Con la clase "DatosArchivos", se procesa el contenido de la base de datos. Primero obtiene la ruta de los documentos y los nombres de estos, los lee, convierte el contenido a minúsculas y elimina los caracteres que dificultan la búsqueda, separando así cada palabra teniendo en cuenta los espacios en blanco. También se encuentran datos como:
  - Cantidad de documentos en la base de datos
  - La frecuencia de cada palabra (cantidad de veces que se repite).
  - Palabra con mayor frecuencia.
  - Cantidad de documentos donde aparece cada palabra
- IDF – TF: Con la clase "CarpetaDeDatos", pasamos al cálculo de IDF – TF de cada palabra. Los resultados serán guardados en diccionarios (IDFS – TFS), otorgando cada valor a la palabra en cuestión.
  - El valor IDF está dado por la fórmula:

$$IDF = log_{10}\frac{TD+1}{CD+1}$$

    Donde TD es el total de documentos existentes en la base de datos, y CD la cantidad de documentos donde aparece la palabra en cuestión.
  - El valor TF está dado por la fórmula:

$$\frac{F}{maxF}$$

    Donde F es la frecuencia de la palabra en cuestión, y maxF la cantidad de documentos donde aparece.
- Motor: El la clase "Motor" desarrollamos el modelo vectorial encontrando el "peso" de cada palabra dado por la fórmula:

$$\frac{(IDF * TF)^2}{IDF^2 * TF^2}$$

Hallara y comparara estos datos antes obtenidos con la "consulta" insertada por el Usuario. Devolviendo así un fragmento de cada texto con la consulta insertada.

- Métodos utilizados: En la clase "Útiles" se desarrollan varios métodos para la facilitación de los procesos anteriores. Estos son:
  - ConsultaSinOperadores: Si la consulta tiene alguno de los operadores especificados, devuelve una lista de arrays con la(s) palabra(s) que contiene el operador junto a este.
  - LimpiarTexto: Este método elimina loa caracteres incomodos a la hora de leer los textos.
  - Encuentra: Busca entre todas las palabras en los datos la que mayor índice de coincidencias tenga.
  - MaximoIndiceDeCoincidencias: Método que devuelve el índice máximo de coincidencias entre dos palabras.
  - Direccion: Obtiene la dirección desde donde se ejecuta la aplicación.
  - ArchivosEnCarpeta Método que busca los archivos.
  - ExtraerPalabras: Extrae las palabras de la consulta en minúscula.
  - ConsultaValida Analiza y valida la consulta.
  - PalabrasSinRepetir: Método para no repetir las palabras a procesar