

Moogle! es una aplicación "totalmente original" cuyo propósito es buscar inteligentemente un texto en un conjunto de documentos. Es una aplicación web, desarrollada con tecnología .NET Core 6.0, específicamente usando Blazor como "framework" web para la interfaz gráfica, y en el lenguaje de visual studio code. La aplicación está dividida en dos componentes fundamentales:

- 'MoogleServer' es un servidor web que renderiza la interfaz gráfica y sirve los resultados.
- 'MoogleEngine' es una biblioteca de clases donde está... ehem... casi implementada la lógica del algoritmo de búsqueda.

En primer lugar, el usuario puede buscar no solo una palabra sino en general una frase cualquiera. - Si no aparecen todas las palabras de la frase en un documento, pero al menos aparecen algunas, este documento también queremos que sea devuelto, pero con un (score) menor mientras menos palabras aparezcan. - El orden en que aparezcan en el documento los términos del (query) en general no debe importar, ni siquiera que aparezcan en lugares totalmente diferentes del documento.

- Si en diferentes documentos aparecen la misma cantidad de palabras de la consulta, (por ejemplo, 2 de las 3 palabras de la consulta "algoritmos de ordenación"), pero uno de ellos contiene una palabra más rara (por ejemplo, "ordenación" es más rara que "algoritmos" porque aparece en menos documentos), el documento con palabras más raras debe tener un "score" más alto, porque es una respuesta más específica. - De la misma forma, si un documento tiene más términos de la consulta que otro, en general debería tener un "score" más alto (a menos que sean términos menos relevantes). - Algunas palabras excesivamente comunes como las preposiciones, conjunciones, etc., deberían ser ignoradas por completo ya que aparecerán en la inmensa mayoría de los documentos (esto queremos que se haga de forma automática, o sea, que no haya una lista cableada de palabras a ignorar, sino que se computen de los documentos).

La idea original del proyecto es buscar en un conjunto de archivos de texto (con extensión '.txt') que estén en la carpeta 'Content' Lo primero que tendrás que hacer para poder trabajar en este proyecto es instalar .NET Core 6.0 . Luego, solo te debes parar en la carpeta del proyecto y ejecutar en la terminal de Linux: