



Moogle!

Primer Proyecto de Programación

Joel Aparicio Tamayo

Facultad de Matemática y Computación
Universidad de la Habana

2023

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

**Motor de
búsqueda**

Funcionamiento

Ver código...

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

¿Qué es Moogle!?

Moogle! es una aplicación web cuyo propósito es buscar inteligentemente un texto en un conjunto de documentos.

Está desarrollada con tecnología .NET Core 7.0, usando Blazor como *framework* web para la interfaz gráfica, y en el lenguaje C-Sharp.

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

¿Qué es Moogle!?

Moogle! es una aplicación web cuyo propósito es buscar inteligentemente un texto en un conjunto de documentos.

Está desarrollada con tecnología .NET Core 7.0, usando Blazor como **framework** web para la interfaz gráfica, y en el lenguaje C-Sharp.

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 implementada la lógica del algoritmo de búsqueda.

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ El proyecto ha sido probado con un repositorio de alrededor de 15 mil documentos cuyo peso era de 170 MB. Con estos documentos el proyecto demora alrededor de 1 minuto inicialmente para cargar, y la búsqueda es casi instantánea.

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ El proyecto ha sido probado con un repositorio de alrededor de 15 mil documentos cuyo peso era de 170 MB. Con estos documentos el proyecto demora alrededor de 1 minuto inicialmente para cargar, y la búsqueda es casi instantánea.
- ▶ La búsqueda muestra los 10 primeros resultados como máximo. (Pudieran ser más si hay scores repetidos)

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ El proyecto ha sido probado con un repositorio de alrededor de 15 mil documentos cuyo peso era de 170 MB. Con estos documentos el proyecto demora alrededor de 1 minuto inicialmente para cargar, y la búsqueda es casi instantánea.
- ▶ La búsqueda muestra los 10 primeros resultados como máximo. (Pudieran ser más si hay scores repetidos)
- ▶ El buscador funciona tanto al seleccionar el botón “Buscar”, como al presionar la tecla “Enter”.

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

**Motor de
búsqueda**

Funcionamiento

Ver código...

Particularidades

- ▶ El proyecto ha sido probado con un repositorio de alrededor de 15 mil documentos cuyo peso era de 170 MB. Con estos documentos el proyecto demora alrededor de 1 minuto inicialmente para cargar, y la búsqueda es casi instantánea.
- ▶ La búsqueda muestra los 10 primeros resultados como máximo. (Pudieran ser más si hay scores repetidos)
- ▶ El buscador funciona tanto al seleccionar el botón “Buscar”, como al presionar la tecla “Enter”.
- ▶ El snippet mostrado en pantalla de cada documento muestra una vecindad de la primera aparición de la palabra de más peso de la query con respecto a dicho documento.

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ Se ha implementado el operador de relevancia $*$, el operador de obligatoriedad \wedge , el operador de inexistencia $!$ y el operador de cercanía \sim .

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ Se ha implementado el operador de relevancia $*$, el operador de obligatoriedad \wedge , el operador de inexistencia $!$ y el operador de cercanía \sim .
- ▶ Los operadores \wedge y $!$ funcionan solo para la primera palabra de la query que los tenga. (Ej: “!hola !Claudia” solo funcionará para “hola”, mientras que a “Claudia” la tratará como una palabra normal)

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle?!

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ Se ha implementado el operador de relevancia $*$, el operador de obligatoriedad \wedge , el operador de inexistencia $!$ y el operador de cercanía \sim .
- ▶ Los operadores \wedge y $!$ funcionan solo para la primera palabra de la query que los tenga. (Ej: “!hola !Claudia” solo funcionará para “hola”, mientras que a “Claudia” la tratará como una palabra normal)
- ▶ Pueden existir combinaciones de los operadores. (Uno en cada palabra distinta, no puede haber dos juntos en una misma palabra)

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle?!

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Particularidades

- ▶ Se ha implementado el operador de relevancia $*$, el operador de obligatoriedad \wedge , el operador de inexistencia $!$ y el operador de cercanía \sim .
- ▶ Los operadores \wedge y $!$ funcionan solo para la primera palabra de la query que los tenga. (Ej: “!hola !Claudia” solo funcionará para “hola”, mientras que a “Claudia” la tratará como una palabra normal)
- ▶ Pueden existir combinaciones de los operadores. (Uno en cada palabra distinta, no puede haber dos juntos en una misma palabra)
- ▶ El operador de cercanía se debe usar de la forma: palabra1 \sim palabra2 (espacio entre el operador y las palabras).

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

Mooglee Server

Es donde se ha implementado la interfaz gráfica de la aplicación. En esta parte se ha agregado una línea en el archivo *Program* para cargar el repositorio antes de ejecutar la aplicación.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

Mooglee Server

Es donde se ha implementado la interfaz gráfica de la aplicación. En esta parte se ha agregado una línea en el archivo *Program* para cargar el repositorio antes de ejecutar la aplicación.

```
26 MoogleeEngine.Repository.CreateRepository();  
27 app.Run();
```

Importante: El proyecto se ejecuta introduciendo el comando: `dotnet watch run --project MoogleeServer` en la terminal, desde la carpeta raíz del programa.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

Mooglee Server

Para el funcionamiento de la búsqueda con la tecla **Enter** se han agregado varias líneas de código en el archivo *Index.razor* de la carpeta *Pages*.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

Mooglee Server

Para el funcionamiento de la búsqueda con la tecla **Enter** se han agregado varias líneas de código en el archivo *Index.razor* de la carpeta *Pages*.

En la línea 7:

```
7 <input class="p-1 my-4" @bind="query" placeholder="Introduzca su búsqueda" @keyup = "@Enter">
```

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

**Motor de
búsqueda**

Funcionamiento

Ver código...

Mooglee Server

Para el funcionamiento de la búsqueda con la tecla **Enter** se han agregado varias líneas de código en el archivo *Index.razor* de la carpeta *Pages*.

En la línea 7:

```
7 <input class="p-1 my-4" @bind="query" placeholder="Introduzca su búsqueda" @onkeyup = "@Enter">
```

En la línea 32:

```
32 private void Enter(KeyboardEventArgs Event) {  
33     if (Event.Code == "Enter") RunQuery();  
34 }
```

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

- *Repository*: Es la clase que genera el repositorio de documentos.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

- ▶ *Repository*: Es la clase que genera el repositorio de documentos.
- ▶ *Document*: Es la clase que tiene los métodos para trabajar con los documentos.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

- ▶ *Repository*: Es la clase que genera el repositorio de documentos.
- ▶ *Document*: Es la clase que tiene los métodos para trabajar con los documentos.
- ▶ *RankingVector*: Es la clase que contiene el método para asignarle un score a cada documento.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

- ▶ *Repository*: Es la clase que genera el repositorio de documentos.
- ▶ *Document*: Es la clase que tiene los métodos para trabajar con los documentos.
- ▶ *RankingVector*: Es la clase que contiene el método para asignarle un score a cada documento.
- ▶ *Operators*: Es la clase que tiene los métodos de los operadores implementados.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

MoogleeEngine

Está compuesta por 8 clases. De ellas hay dos que ya estaban creadas y que no fueron modificadas, que son: *SearchItem* y *Searchresult*.

- ▶ *Repository*: Es la clase que genera el repositorio de documentos.
- ▶ *Document*: Es la clase que tiene los métodos para trabajar con los documentos.
- ▶ *RankingVector*: Es la clase que contiene el método para asignarle un score a cada documento.
- ▶ *Operators*: Es la clase que tiene los métodos de los operadores implementados.
- ▶ *Mooglee*: Es la clase principal del proyecto y donde se generan los resultados de las búsquedas.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de búsqueda

Funcionamiento

Ver código...

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de búsqueda

Funcionamiento

Ver código...

Funcionamiento

El funcionamiento en tiempo de búsqueda del programa está regido por la clase Mooglee. En esta clase se recibe la query como parámetro del método Query y se trabaja con ella.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

Motor de búsqueda

Funcionamiento
Ver código...

Funcionamiento

El funcionamiento en tiempo de búsqueda del programa está regido por la clase Mooglee. En esta clase se recibe la query como parámetro del método Query y se trabaja con ella.

Primeramente, se guarda la query separada por palabras. Guardamos además en variables auxiliares los valores que resultan de evaluar cada método de la clase Operators. También creamos un array para guardar los valores de IDF de cada palabra de la query, un diccionario para relacionar más adelante los scores con el documento correspondiente, un diccionario para relacionar cada palabra de la query y la cantidad de veces que se repite en la misma.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Funcionamiento

Luego con un foreach llenamos el diccionario poniendo cuántas veces se repite cada palabra en la query. En un segundo momento, se crea un primer bucle for para calcular los IDF de las palabras de la query. Para ello se utiliza una condicional, de manera que, si la palabra está en algún documento, entonces calculamos su IDF y lo multiplicamos por su valor de relevancia y también por la cantidad de veces que está la palabra en la query. En caso contrario pues mantenemos el cero.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Funcionamiento

Luego de tener los valores de cada palabra de la query, entonces se crea un array donde se va a guardar el ranking de los documentos utilizando el método `GetScore` de la clase *RankingVector*. Después, con un pequeño bucle `for`, le asociamos a cada valor almacenado en `score` el documento asociado a él (puede haber varios con un mismo valor de `score`, por eso el diccionario recibe como valor una lista). Luego ordenamos el ranking de mayor a menor.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Funcionamiento

Luego de tener los valores de cada palabra de la query, entonces se crea un array donde se va a guardar el ranking de los documentos utilizando el método `GetScore` de la clase *RankingVector*. Después, con un pequeño bucle `for`, le asociamos a cada valor almacenado en `score` el documento asociado a él (puede haber varios con un mismo valor de `score`, por eso el diccionario recibe como valor una lista). Luego ordenamos el ranking de mayor a menor.

Una vez ordenado el ranking, entonces se crea una lista de objetos `SearchItem` (título, snippet, score) que representan cada documento a mostrar en pantalla.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee!?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Funcionamiento

Luego existe una condición por si no existen resultados de búsqueda:

Moog!e!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moog!e!?
Particularidades

Estructura

Moog!e Server
Moog!eEngine

**Motor de
búsqueda**

Funcionamiento
Ver código...

Funcionamiento

Luego existe una condición por si no existen resultados de búsqueda:

```
807
81  /* Si el mayor score posee ese valor significa que no hay resultados, pues se restó 10* words.Length
82     cada vez que no se encontró una palabra de la query. O bien si todo es 0, es porque es una palabra
83     que está en todos los documentos y no nos interesa*/
84     if ( (score[0] == -10 * wordsQuery.Length * wordsQuery.Length) || ((score[0] == score[score.Length -1]) && score[0] == 0) )
85     {
86         items.Add (new SearchItem("No hay resultados", "El repositorio no contiene la búsqueda deseada", 0f));
87         return new SearchResult(items.ToArray(), query);
88     }
89
```



¿Quisiste decir etpdmej?

• No hay resultados

... El repositorio no contiene la búsqueda deseada ...

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?

Particularidades

Estructura

Moogle Server

MoogleEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Funcionamiento

En caso de que sí existan resultados, entonces eliminamos primero los valores repetidos en score, y luego se procede a mostrar los diez documentos con más coincidencias con la búsqueda realizada por el usuario.

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

**Motor de
búsqueda**

Funcionamiento

Ver código...

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogles!?

Particularidades

Estructura

MoogLe Server

MoogLeEngine

Motor de búsqueda

Funcionamiento

[Ver código...](#)



- King, Stephen - Historias fantasticas

• **Piedra de nieblas**

Funcionamiento

En caso de no existir resultados, se le añade lo siguiente al código y se muestra en pantalla:

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?
Particularidades

Estructura

Mooglee Server
MoogleeEngine

**Motor de
búsqueda**

Funcionamiento
Ver código...

Funcionamiento

En caso de no existir resultados, se le añade lo siguiente al código y se muestra en pantalla:

```
115     /* Si "items" está vacío es porque se emplearon operadores y ningún documento cumple con las
116     especificidades. */
117     if (items.Count() == 0) items?.Add(new SearchItem("No hay resultados", "Pruebe eliminar las especificidades", 0F));
118     return new SearchResult(items?.ToArray(), query);
119 }
```



novela *wonzxm

Buscar

¿Quisiste decir [novela wonxwm](#)?

• No hay resultados

... Pruebe eliminar las especificidades ...

Mooglee!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Mooglee?

Particularidades

Estructura

Mooglee Server

MoogleeEngine

Motor de
búsqueda

Funcionamiento

Ver código...

Ver código...

Link de la carpeta MoogleEngine:

<https://github.com/Joel0347/Moogle-/tree/main/Moogle!/MoogleEngine>

Link de la carpeta MoogleServer:

<https://github.com/Joel0347/Moogle-/tree/main/Moogle!/MoogleServer>

Moogle!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moogle!?
Particularidades

Estructura

Moogle Server
MoogleEngine

Motor de búsqueda

Funcionamiento
Ver código...

Moog!e!

Joel Aparicio
Tamayo C-121

Introducción

¿Qué es Moog!e?
Particularidades

Estructura

Moog!e Server
Moog!eEngine

Motor de búsqueda

Funcionamiento
Ver código...

Moog!e!

Primer Proyecto de Programación

Joel Aparicio Tamayo

Facultad de Matemática y Computación
Universidad de la Habana

2023