



## Proyecto de programacion

Adrián Alejandro Souto Morales

25 de julio de 2023

C-122

# MoogleServer/Program.cs

En *MoogleServer/Program.cs* se llama a la función **GetData()** que carga todos los datos necesarios que se ejecuta mientras se monta el servidor:

```
MoogleEngine.LeerDocs.GetData();  
app.Run();
```

Figura: Fragmento de MoogleServer.cs

La clase estática “**LeerDocs**” contiene la función “**GetData()**” para crear un “**Vector**” por cada documento con las respectivas palabras que lo componen.

```
string[] files = Directory.GetFiles(path, "*.txt");
foreach(string file in files){
    string text = "."+File.ReadAllText(file).ToLower()+ ".";
    //Creo una lista con todas las palabras del texto
    List<string> listWords = (Utils.MakeList(text).ToList());
    //Creo un Vector por cada documento (Al crearse se guarda en mi matriz de docs)
    Vector v = new Vector(
        file,
        listWords
    );
}
```

Figura: Función de GetData()

# Vector

## Propiedades:

- **TFIDF** (Dictionary<string, double>)
- **Path** (string)
- **Words** (List<string>)

## Métodos:

- **CountWords()**
- **GetName()**
- **ProdEscalar(Vector v)**

# Matriz

## Propiedades:

- **MatrizVectores**(List< *Vector*>)
- **IDF**(Dictionary<*string*, *int*>)

## Métodos:

- **Add**(Vector v)
- **CalculateIDF**(string w)
- **CalculateTF**(Vector v, string w)

La función **Search()** ubicada en la clase `Moogles.cs` retorna un array de **SearchItem**:

- 1 **Calcula el score** por cada vector y solo lo muestra si es mayor a  $10^{-6}$  (Producto escalar entre el query y el vector)
- 2 Guarda la posición de la palabra con más TF-IDF del query que esté en el documento
- 3 **Guarda el snippet** dado por un fragmento del texto que contenga dicha palabra (Desde punto (“.”) anterior a 50 caracteres antes de la palabra hasta el siguiente punto después de 100 caracteres) Para ello se usan las funciones `NextDot()` y `PrevDot()` de mi clase `Utils`.
- 4 Guarda en una Lista de `SearchItem` los datos del vector (nombre, snippet, score)
- 5 Cuando termina de recorrer los vectores ordena la Lista de `SearchItem` según su score, la convierte en Array y la devuelve

## Métodos:

- **MakeList**(string text)
- **PrevDot**(int pos, string text)
- **NextDot**(int pos, string text)