**Reporte de la Actividad Integral 3.4**

Para la solución de este problema decidimos usar C++, un lenguaje con el que todos en el equipo tenemos experiencia. Para evitar usar arreglos de dos dimensiones por cada nodo de la Maquina de Estados, usamos la librería std::map. Esta librería te permite almacenar un dato con una clave única. Cada dato tiene una clave única e irrepetible que ayuda a su búsqueda. Las llaves se ordenan con una función dentro de la librería, lo que reduce en gran medida la complejidad al momento de buscar un dato.

En el caso de nuestro proyecto, hicimos un mapa que almacenaba todos los tipos de carácter que nos podíamos encontrar, desde los operadores matemáticos hasta los operadores lógicos. También, hicimos un mapa por cada uno de los nodos de nuestra Maquina de Estados, que incluye el estado o estados que acepta ese nodo, además de con que otros nodos está conectado. Con esas dos herramientas, lo que realiza nuestro algoritmo es lo siguiente:

1. La lectura del archivo “.txt” y el almacenamiento de los datos en un vector.
2. El lector entra al leer y es comparado con el mapa de caracteres para después entrar en la Maquina de Estados.
3. La Maquina de Estados regresa el estado en el que se encuentra y determina el tipo de token del carácter que se está evaluando.
4. Se imprime en la terminal el Token, en el que cada Token tiene un código especial de color en formato de HTML para poderlo visualizar en un web browser.

Para poder calcular la complejidad de los algoritmos que realizamos tenemos que tomar en cuenta la parte mas importante en nuestro código: los mapas. La complejidad de esta estructura de datos es de O(logn). Las operaciones que realiza la Maquina de Estados, al ser puramente lógicos, no le agregan complejidad a la solución, por lo que la complejidad final del programa seria O(logn). Esto se puede comprobar con el tiempo de ejecución, ya que este es de solamente 1 segundo.

En conclusión, los algoritmos creados en la solución para la Actividad Integral son eficientes en cuestión de complejidad y de tiempo de ejecución. Esto se comprueba con el tiempo de ejecución y la complejidad de la estructura de datos que se utilizó en el programa. El tener una solución eficiente es uno de los objetivos principales de la Ingeniería en Software, para reducir costos y recursos del proyecto.