



# Tecnológico de Monterrey

## **Actividad Integradora 3.4 Resaltador de sintaxis**

Eugenio Andrés Mejía Fanjón A01412143

Fernando Daniel Monroy Sánchez - A01750536

TC2037: Implementación de Métodos Computacionales

Docente: Jesús Guillermo Falcón Cardona

13 de mayo del 2024

## Complejidad del Algoritmo

La complejidad del algoritmo depende de varios factores. Primero, se itera a través de cada línea del archivo y, dentro de cada línea, se itera sobre cada patrón de token. Esto resulta en una complejidad básica de  $O(n * m)$ , donde  $n$  es el número de líneas y  $m$  es el número de patrones.

Para cada patrón, se utilizan expresiones regulares para identificar todos los matches que hacen overlapping. La complejidad de búsqueda con una expresión regular puede variar, pero generalmente es  $O(k)$  por línea, siendo  $k$  la longitud de la línea.

En cada match, se realizan manipulaciones de strings, como la inserción de etiquetas HTML, que generalmente tienen una complejidad de  $O(p)$ , donde  $p$  es el tamaño del string a manipular.

Considerando todos estos factores, la complejidad total aproximada del algoritmo sería  $O(n * m * k * p)$ , lo cual puede ser considerablemente alto, especialmente para archivos grandes con muchas líneas y tokens. Como sabemos que tenemos un número de tokens constante, se podría decir que es  $O(10 * n * k * p)$ .

## Reflexión

En este proyecto, desarrollamos un resaltador de sintaxis para código Python utilizando expresiones regulares. Nuestro objetivo era mejorar la legibilidad del código al clasificar y resaltar diferentes tokens con etiquetas HTML. Las expresiones regulares resultaron ser muy potentes para identificar patrones del texto, a partir de la gramática de la documentación de Python.

Comprendimos que una gramática bien definida es crucial para que un lenguaje de programación sea efectivo y para que las expresiones regulares puedan parsear el código correctamente. Esto asegura que cada token sea identificado y clasificado adecuadamente.

Si bien el proyecto fue un ejercicio valioso, nos dimos cuenta de que la eficiencia del algoritmo es clave para su uso práctico. Para mejorar la herramienta, sería útil optimizar las expresiones regulares y considerar algoritmos de análisis más eficientes, reduciendo así la complejidad computacional y mejorando su rendimiento en entornos de desarrollo reales.