

La elaboración de este resaltador de sintaxis se hace por medio de distintas funciones en racket usando expresiones regulares. Hubiera sido mucho menos eficiente el programa en caso de no contar con el uso de listas para recorrer cada una de las líneas del código de testing, estas listas a su vez permiten y facilitan hacer las comparaciones por medio de ciclos. Los ciclos son muchos más eficientes ya que al utilizar las expresiones regulares se busca un patrón en lo que le estas pasando, es decir simplifican y automatizan el análisis de las líneas para su clasificación por medio de clases. Estos elementos nos ayudan a ahorrar tiempo en la ejecución del programa, evitan líneas innecesarias y aumentan la productividad. Al usar Racket y ser un lenguaje funcional permite mucha flexibilidad, aparte tiene muchas bibliotecas que facilitaron la elaboración del programa al reducir las funciones que teníamos que implementar. La complejidad de este programa en términos de tiempo de ejecución depende del tamaño del archivo que se está procesando. En general, la complejidad del programa es $O(N*M)$, donde N es el número de líneas en el archivo y M es la cantidad de expresiones regulares en la lista classes. Esto se debe a que se realiza una búsqueda de expresiones regulares para cada línea del archivo, y el tiempo de ejecución de la búsqueda de expresiones regulares es proporcional a la cantidad de expresiones regulares en la lista classes. Además, la función pack que se llama después de la clasificación tiene una complejidad de $O(N)$, donde N es el número de elementos en la lista content.