



**Tecnológico
de Monterrey**

TC2037.601

Actividad Integradora 3.4 Reflexión

Erwin Porras Guerra A01734881

23/04/22

Para realizar esta actividad integradora utilizamos varios métodos y lenguajes de programación, los principales de estos fueron Elixir y Erlang. Erlang se encargaba de analizar la entrada y asignarle un “token” dependiendo de su categoría léxica, para Python, que es el lenguaje que quisimos analizar, estas categorías fueron:

- Literales Enteros
- Literales Flotantes
- Strings
- Operadores
- Keywords (palabras clave)
- Números Imaginarios
- Delimitadores
- Espacios
- Comentarios
- Nueva Línea

El siguiente paso era utilizar las tokens previamente asignadas y a base de ellas asignarles los tags de HTML correspondientes para poder darle formato a cada elemento de la entrada, este proceso se llevo a cabo utilizando el lenguaje de programación Elixir. Finalmente, con un script de CSS se le asignó el color a cada elemento. Las categorías léxicas que dejamos fuera son:

- Literales Bytes
- Indentación Propia
- Strings Formateadas
- Unión de Línea Explícita
- Unión de Línea Implícita

Dejamos estas categorías fuera porque el implementarlas requerían técnicas más avanzadas de programación y asignación de memoria que estaban fuera del alcance de este avance. Para literales bytes y strings formateadas, en cuanto al proceso de resaltamiento léxico, estas se podían considerar igual que strings ya que al prefijar estas con sus caracteres correspondientes se comportaban igual; es importante aclarar que esto solo es durante el resaltamiento léxico, ya que ante el lenguaje Python y su compilación, estas si tienen diferentes usos, funciones y comportamientos. En cuanto al impacto ético, este programa genera un impacto positivo, aunque sea pequeño, ya que me permitió a mi aprender más acerca de los lenguajes aplicados, además de que podemos entender la forma la que trabajan algunos editores de código en los que, de acuerdo con el lenguaje utilizado, los colores de las palabras, operadores y palabras claves pueden tener un color en específico. Además, podemos utilizarlo para leer un código de algún lenguaje que nos sea desconocido, ya que de esta forma podemos reconocer de forma más sencilla su estructura.

Por último, la complejidad de nuestro algoritmo es lineal, ya que la cantidad de componentes de sintaxis que son introducidos en nuestro lector de expresiones regulares es la misma a la cantidad de elementos que tiene que procesar y clasificar nuestro código de elixir. Podemos afirmar que el tiempo de ejecución de nuestro código es equivalente al calculo que hicimos relacionado con su complejidad lineal cuando utilizamos la herramienta time antes de ejecutarlo.