



**Tecnológico
de Monterrey**

Project 2: Lexical Highlighter

Rogelio Guzman Cruzado A01639914

Fernanda Elizabeth Romo Alarcon A10639261

27/04/2023

Implementación de Métodos Computacionales

1. Selecciona un lenguaje de programación que te resulte familiar (por ejemplo, C, C++, C#, Java, JavaScript, etc), y determina las categorías léxicas (por ejemplo, palabras reservadas, operadores, literales, comentarios, etc.).

Nuestro lenguaje de programación de preferencia sobre el cual trabajaremos todo este proyecto será Python.

En python pudimos identificar estas categorías:

- Clases (tipos de datos (transforman datos, no declaran variables)): int, bool, complex, float, str
- Palabras reservadas: print, if, for, while, private, public, false, true
- Literales de int: 0...n
- Variables: Cualquier cadena de caracteres que cumpla ciertas condiciones.
- Operadores: +, -, =, /, *,
- Comentarios: Cualquier cosa después de un #

2. Diseña las expresiones regulares de las categorías léxicas (mínimo 6).

- Classes: `int|bool|complex|float|str`
- Reserved: `public |private |for |while |if |true |false |if |in |print`
- String literals: `"[^"]*"'`
- Operators: `(\+|\-|*|\/|\%|\=|\+=|\-=|*=|\/=|\%=|**=|\/\=|\/\(|\))`
- Int literals: `\b\d+\b`
- Variables: `[a-zA-Z_][a-zA-Z0-9_]*`
- Comments: `(#.*)`

3. Usando un lenguaje de programación implementa el motor de expresiones regulares

Utilizaremos el lenguaje Python para programar nuestro motor de expresiones, así mismo, el lenguaje sobre el cual queremos resaltar el léxico de sus categorías también será Python. Así que el código que se pedirá en `entrada.txt` será un código de Python.

El código se encuentra en `main.py`.

Liga de replit como alternativa.

<https://replit.com/join/metoizxnrj-ferromo>

4. Explicación de funcionamiento de código

El programa `main.py` leerá un archivo `txt` de nombre *entrada.txt*. Nuestro programa recibe el texto del archivo de entrada línea por línea y los introduce en una lista sobre la cual serán utilizadas nuestras expresiones regulares. Así mismo, nuestro programa crea un nuevo archivo `HTML` sobre el cual se escribirá por default el formato necesario para interpretarse como un archivo `HTML` válido. Al finalizar el proceso de match de las

expresiones regulares a nuestro texto, serán introducidos a nuestro archivo HTML nuevos párrafos y spans con atributos de color distintos dependiendo a qué categoría léxica hizo match la palabra determinada en ese momento, así como la palabra.

Este es el código de colores que utilizamos.

■ Clases	■ Palabras Reservadas	■ String Literals
■ Operadores	■ Int literals:	■ Variables
		■ Comments: (#.*)

El programa automáticamente actualizará el archivo *output.html* cada que se corra el *main.py*, ya que lo elimina y crea uno nuevo en cada run. Debido a esto, puedes editar el archivo de *entrada.txt* las veces que quieras, y se verifica el nuevo archivo cada vez que se corra el main. Cabe mencionar que el lenguaje de programación que se utilizó es python, y que es el mismo que lee y verifica el lexical highlighter.

La librería principal que utilizamos es regex para python, ya que esto nos brinda distintas formas de crear nuestras expresiones regulares fácilmente, y varias opciones para matchear las palabras en nuestro archivo.