

Manual técnico

Juan José Rodas Mansilla



202200389

LENGUAJES FORMALES Y DE PROGRAMACIÓN

Universidad de San Carlos de Guatemala

**Explicación Lógica Del Programa:**

1. Vista:

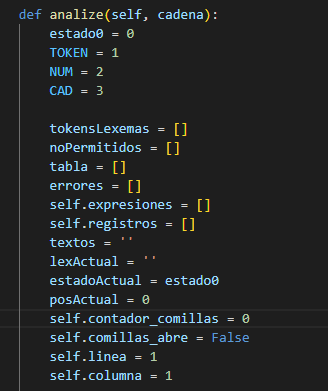
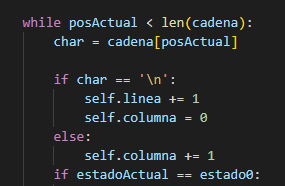
La vista esta hecha de manera que se distribuirá en dos frames, el primero estará posicionado en la parte superior, dicho frame llevara la distribución de los botones que trabajaran en el programa. Haciendo que estos se distribuyan en forma de tabla mediante grid. Por lo que, el ordenamiento que cada botón tendrá será porcentual. El botón encargado de llevar la apertura del txt se encontrará en la parte izquierda del frame, mientras que el botón de traducir se posicionará en la derecha.

Las entradas de texto se encuentran en el segundo frame, el primer cuadro tendrá una barra scroll la cual ayudará para la correcta lectura del archivo. Este primer cuadro podrá ser editado en todo momento aún después de llevar un archivo encima, esto para editar en ultimo cuadro la entrada misma.

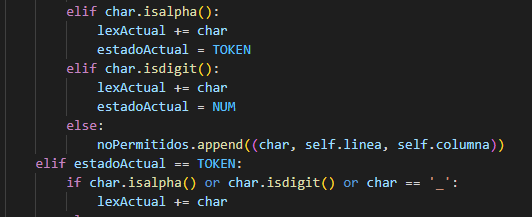
El segundo cuadro únicamente contara con el texto traducido, para tener un mejor control este cuadro no podrá ser editado

Esta estructura se realizo mediante la librería de Python Tkinter, utilizando la mayoría de las herramientas que esta dispone

El analizador lleva una serie de palabras reservadas. Estas se encargaran de llevar los errores léxicos y de identificar cada uno de los lexemas correctos que se han encontrado dentro de la entrada de texto.

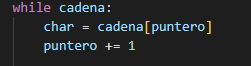
 

El analizador se ira moviendo entre estados dependiendo de lo que encuentre, si es un carácter o si es un digito, inlcuso si es un carácter especial lo tomara en cuenta dependiendo de la posición en la que se encuentre. Cada que se encuentre una palabra especial este tomara un camino diferente, analizando su contenido de foma independiente del resto. Tomando en cuenta que pueden venir varios elementos encerrados en llaves esto es beneficioso para la lógica del programa, ya que ira de registro en registro descartando errores y continuando con la ejecución del programa.

Esto para tomar en cuenta solo las posiblidades habladas en la realización del proyecto.

El analizador léxico llevara toda la entrada de texto recorriéndola una por una dentro de cada carácter. Cada que recorra un carácter aumentara la columna en la que esta, ya que es necesario para el orden de recubrimiento y el listado final de lexemas y de errores.

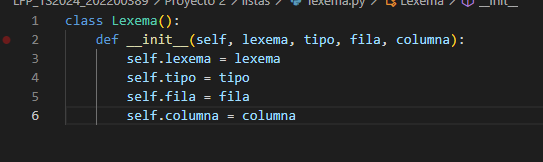


Esto mediante un while y un puntero el cual llevara el listado de caracteres. Luego sumando al puntero y se encuentra una palabra reservada se guarda, caso contrario con un error igualmente se guardará. Al encontrarla saltara a la siguiente línea para continuar con la lectura del texto.

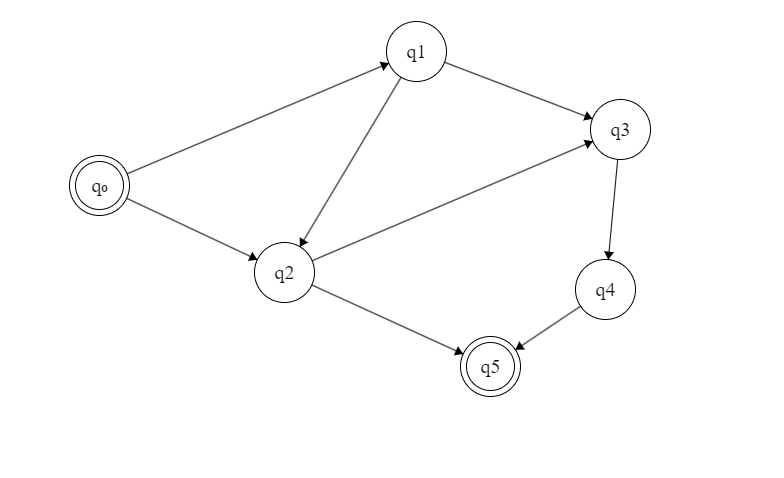
Si se elige la opción 1 del menú se finalizará con el programa, ya que este funciona con un while la condición ya no se cumplirá mediante un Break. Luego mostrara un mensaje en donde confirme la detención del programa.



**Estructuras de la lista de Lexemas o Tokens:**

****

**AFD:**

****

Cada estado tiene una función diferente.

q0 se encargará de analizar la cadena principal del enviado al programa, q1 se referirá a la verificación completa de la cadena en busca de lexemas válidos. Para que posteriormente q2 lleve el control de los errores para posteriormente crear el listado de html con cada uno de los registros encontrados, en caso de que estos vayan vacíos no se mostrara información.

q3 y q4 tiene la sintaxis en cuestión, verificando el orden de las palabras reservadas, haciendo que el programa funciones como se espere, q5 tiene una función parecida, aunque su función llegue de ultimo el analizador lexico estará ayudando desde un principio verificando la llegada de las llaves de apertura, devolviendo advertencias de ser necesario si se encuentra un carácter en una posición no esperada para finalmente devolver las listas de los errores y lexemas encontrados junto con las cadenas encontradas para la verificación del ER.