Diseño de Compiladores   
   
 Facultad de Sistemas   
 Universidad Autónoma de Coahuila

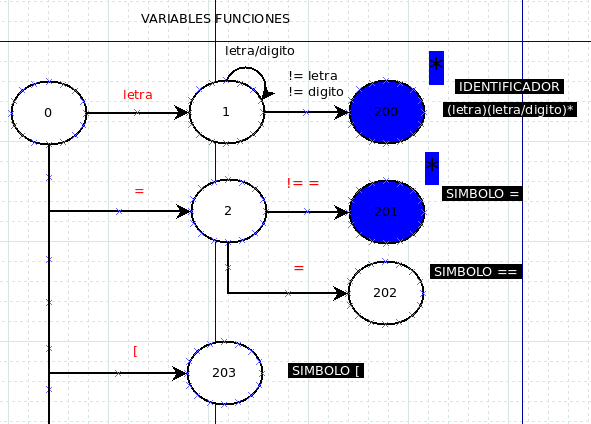
**26/05/2023**

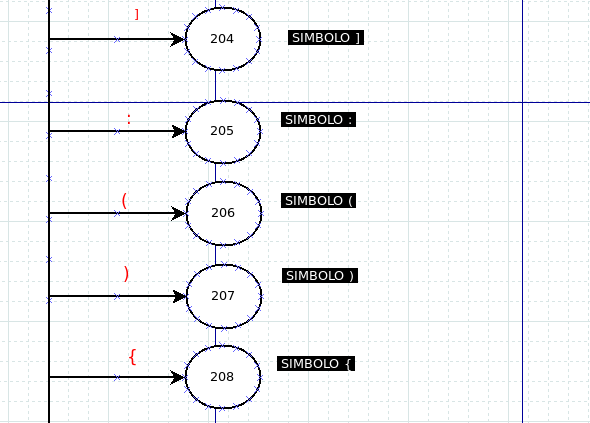
# **Proyecto del Compilador**

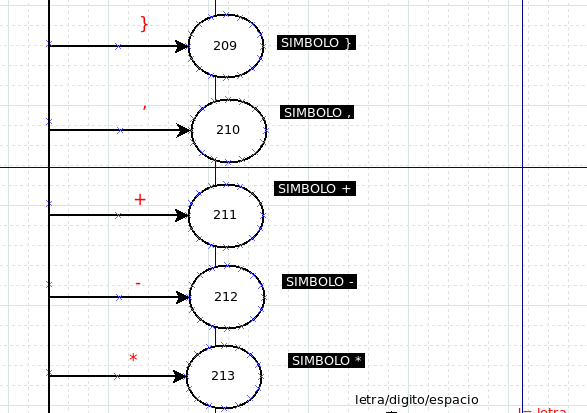
**Diseño del lexico**

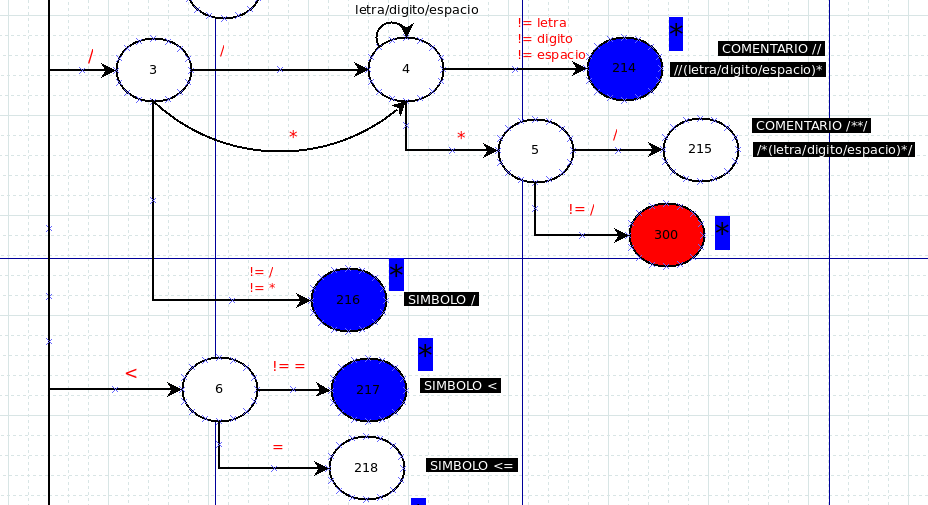
Para el diseño del lexico debemos separar de nuestra gramatica los estados finales y estados intermedios, de los cuales desarrollamos nuestro automata.

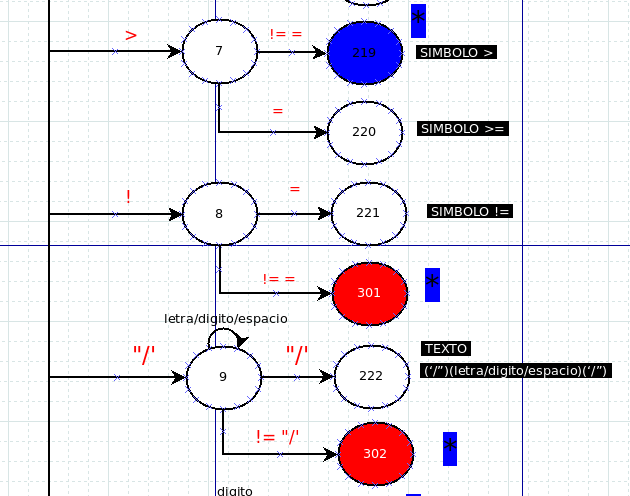
**DISEÑO DEL AUTOMATA (LÉXICO)**

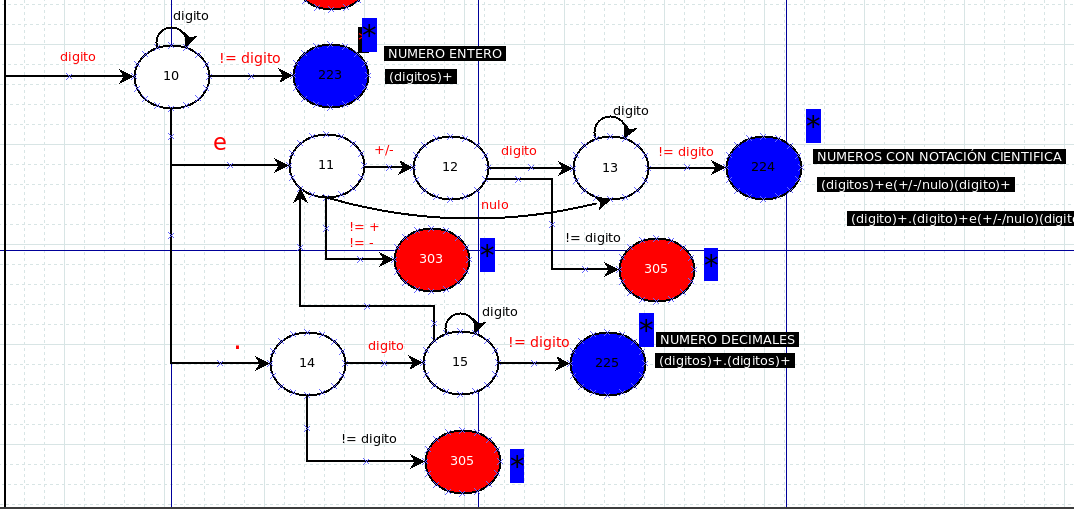
****

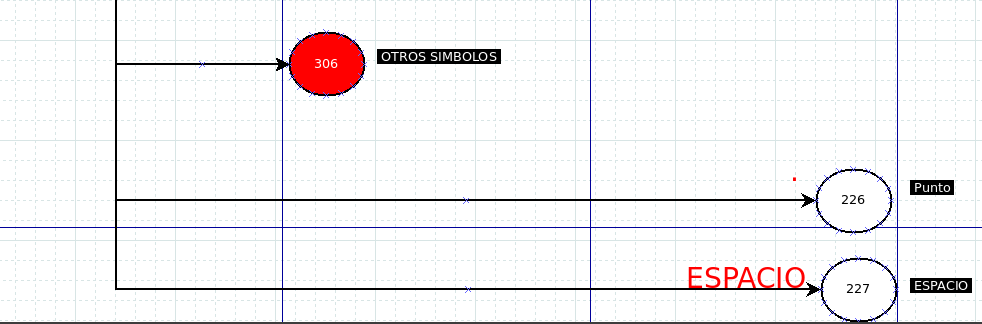
****

****

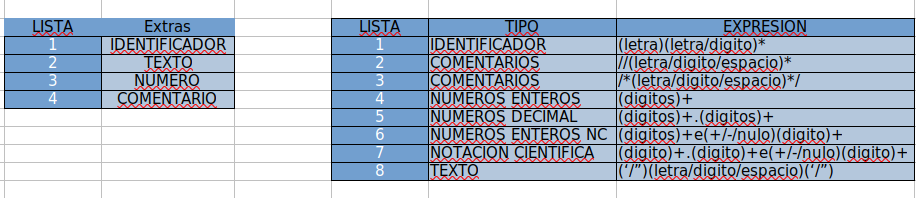
****

****

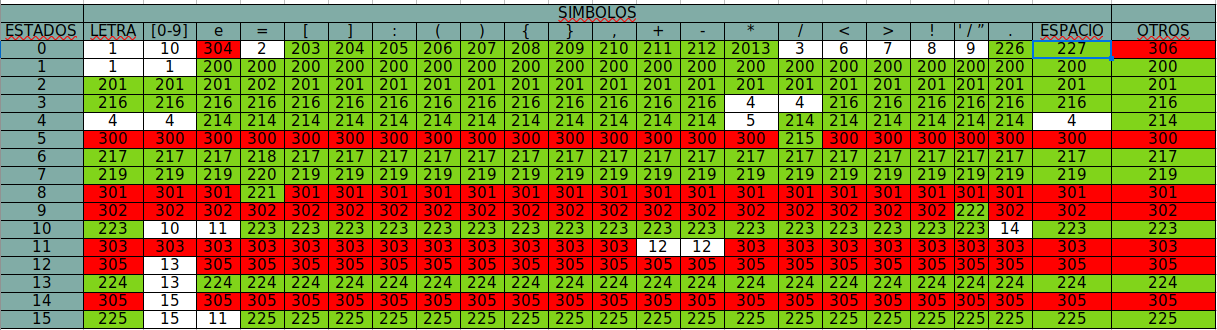
****

****

**IDENTIFICACIÓN**

****

**MATRIZ DE ESTADOS**

**CODIGOS DE ESTADOS (ERRORES Y EXITOS)**

****

**PROGRAMACIÓN**

Tenemos 3 archivos de los cuales uno es el de **lexico.py** contiene una clase llamada **LEXICO** con un constructor que recibe como parametros el nombre del archivo y la mastriz de estados.

Dentro del programa **lexico.py** contiene un metodo llamado **OpenFile()** el cual se encarga de extraer y guardardar en una variables tipo String el contenido del archivo.

Despus tenemos otro metodo de esta clase llamada **LexicoAnalysis()** el cual recibe como parametro el codigo del archivo, dentro de este metodo se crean estas variables.

state = 0 # ESTADOS OBTENIDOS

counter = 0 # APUNTADOR DE CARACTERES

save\_char = [] # GUARDA EN UNA LISTA LOS CARACTERES OBTENIDOS

source\_code += "\n" # SE AGREGA UN SALTO DE LINEA PARA MARCAR FIN DE LECTURA DEL ARCHIVO

size\_code = len(source\_code) # TAMAÑO DEL CÓDIGO FUENTE

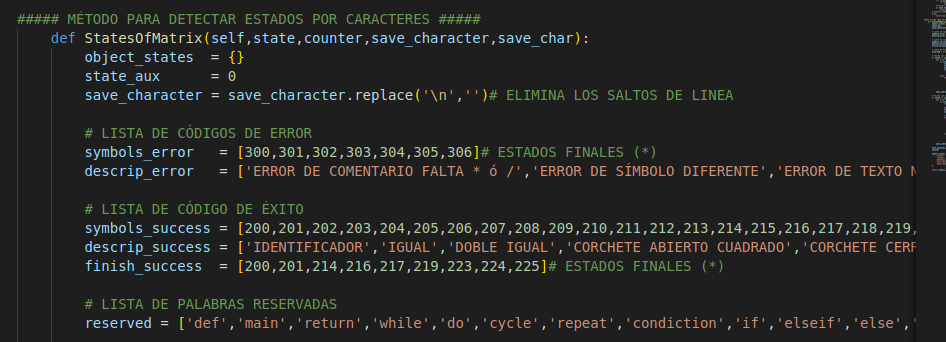
previous\_state = 0 # SE GUARDA EL ESTADO ANTERIOR

save\_character = '' # GUARDA EN UN STRING LA RECOPILACIÓN DE CARACTERES

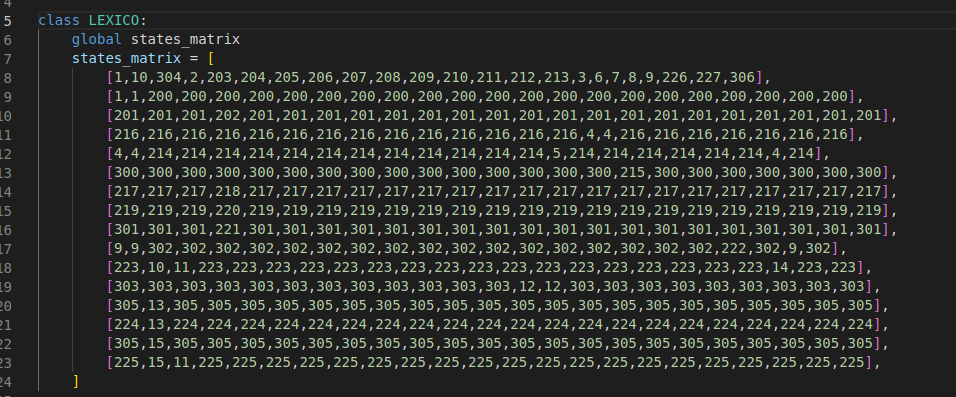
dentro se realiza una instancia de la clase **STATES()** que se encuentra en el archivo **lexico\_state.py,** esta clase contiene el metodo llamado **GetColumn()** donde se obtiene la columna del carácter encontrado y retorna la columna.

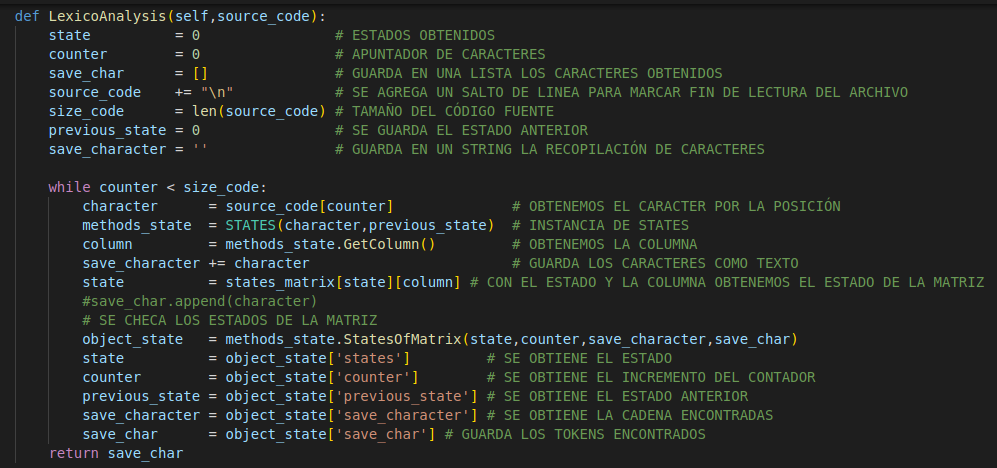
De la siguente metodo llamado **StatesOfMatrix()** donde se detecta el carácter encontrado despues de obtener el estado y columna de la matriz y en este metodo se busca y analiza el estado para identificar el carácter.

Generamos una lista de los estados finales tanto los estados de éxito como los estados de error para identificar si el estado pertenece alguna de esta lista para entrar a la validación y poder agregar el simbolo a la pila de los tokens



Agregamos nuestra matrix de estados a nuestro codigo como una matrix en python.

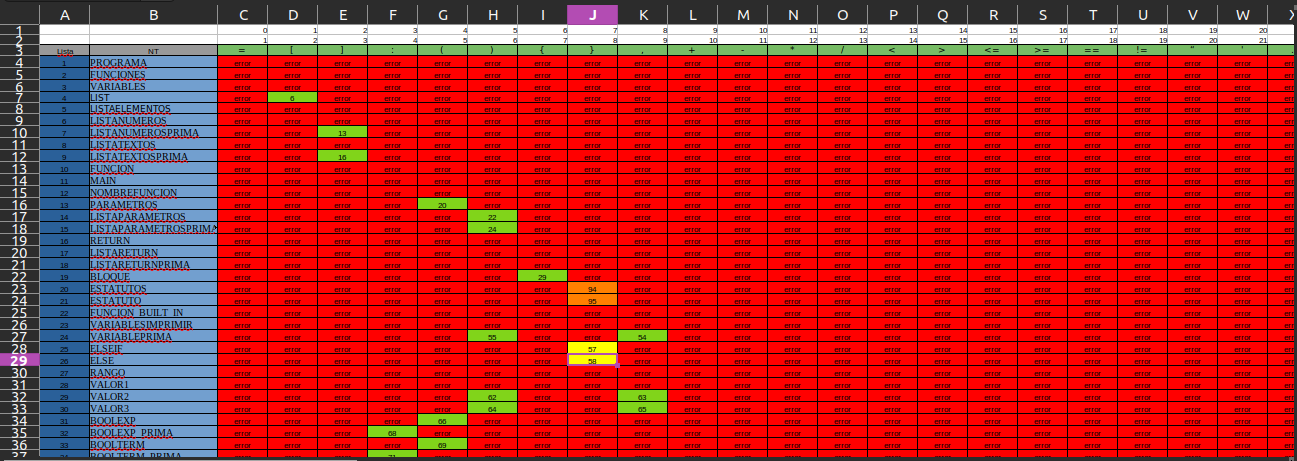
Y generamos el analisis del lexico recolectando nuestro caracteres y agregandolo en los tokens.



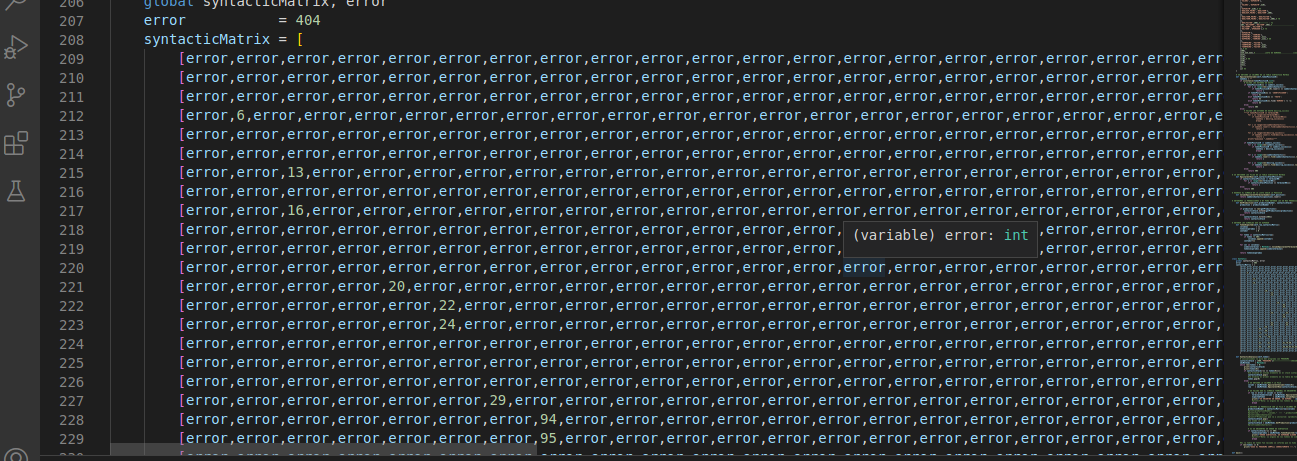
**Diseño de Sintactico**

Generamos una vista de los FIRTS y FOLLOWS de la gramatica las cuales se encuentran en la misma carpeta de este documento.

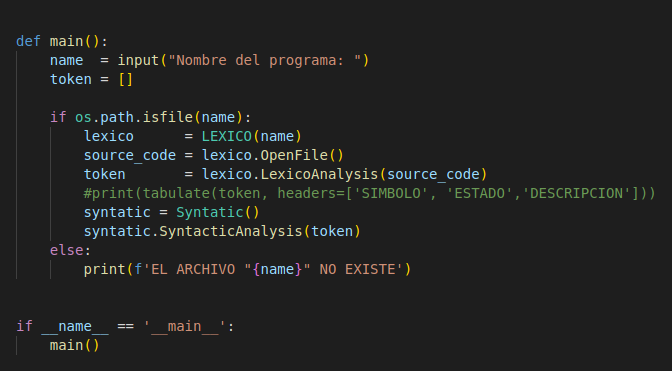
Creamos nuestra matriz apartir de los FIRTS y FOLLOWS como se muestra a continuación.

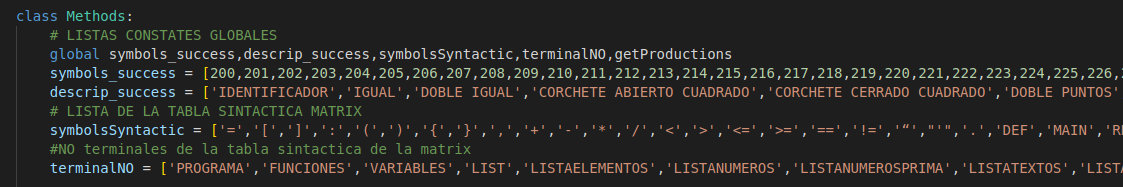
En base a eso comensamos a realizar la programación, se genere un archivo llamado sintactico.py el cual invoca los tokens generados por el lexico para el analisis de nuestra gramatica.

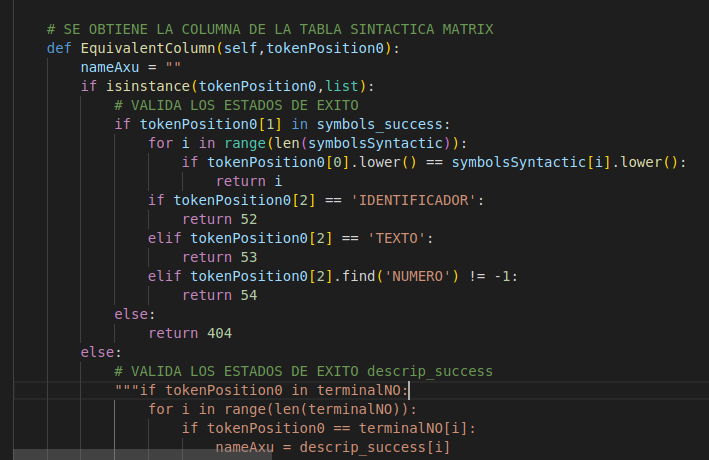
Agregamos nuestra matris del sintactico



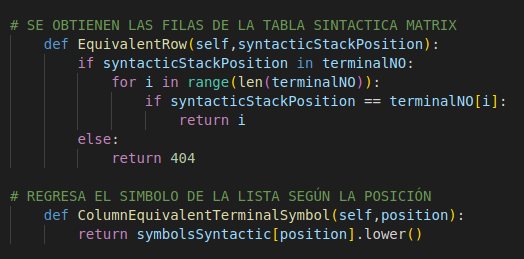
Creamos el objeto del lexico para obtener los tokens en nuestra función main y por terminal mandamos hablar el programa de ejemplo en este caso es el que se llama **hola.txt**

Al igual que el lexico, tambien generamos una lista para los estados y listado de las producciones para la pida del sintactico.

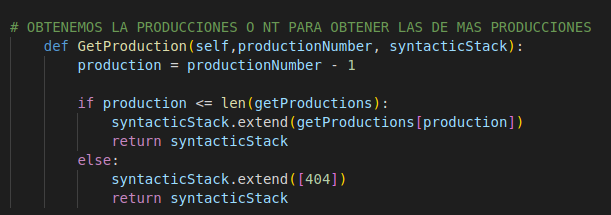
Creamos un metodo que obtenga la columna de la matriz del sintactico.



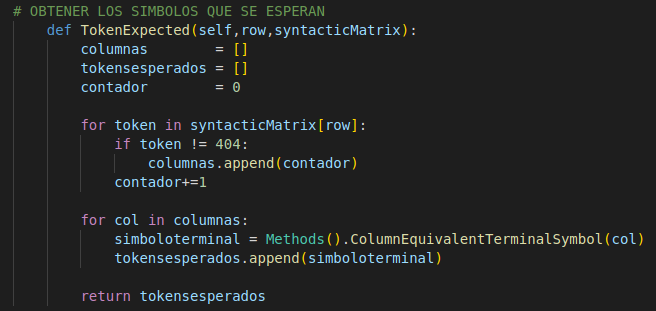
Creamos los siguientes metodos, obtener la fila de la matriz sintactica y obtener el simbolo según la posición que le llega.

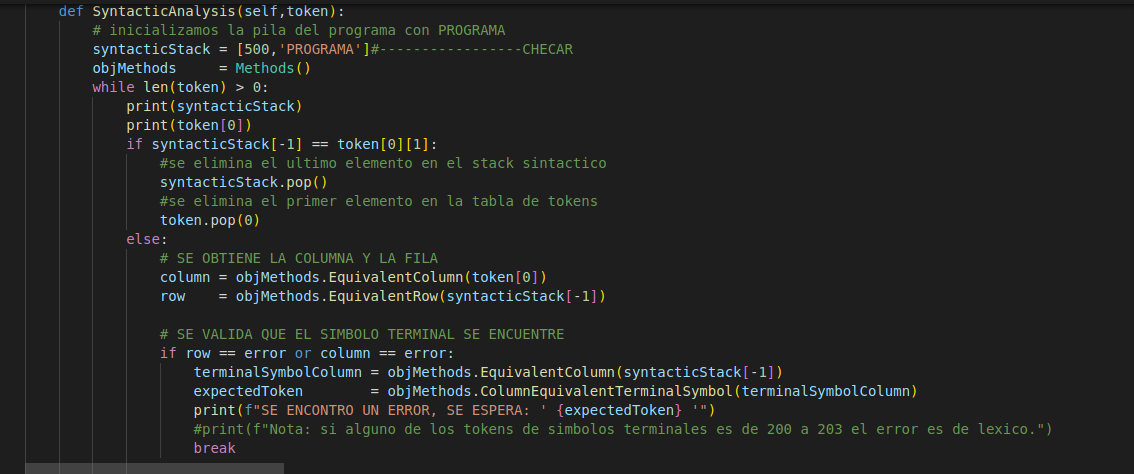
****

Creamos un metodo para obtener las producciones no terminales

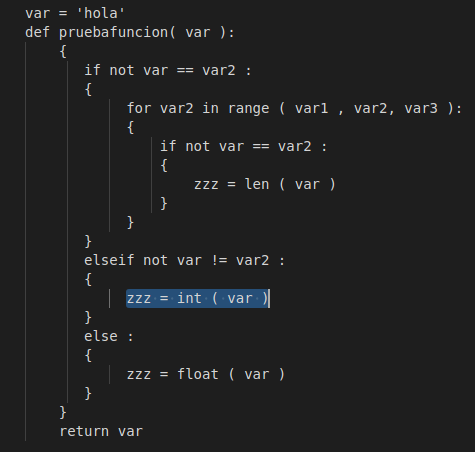


Creamos un metodo con el cual regresaos los tokens esperados al marcar un error sintactico.

Creamos un metodo que analisa y orquesta todos los metods mensionados y valida que la tabla de tokens y la pila de las produciones coincida, si no genera verifica que no haya erro sintactico y de ser posible nos retorna los tokens esperados como en el método anterior.



**ARCHIVO DE PRUEBA: hola.txt**



Al ejecutar el programa y si todo va bien se concluye que el programa compilo correctamente.

