

ANALIZADOR SINTACTICO

OMAR ALEJANDRO QUIROZ TRUJILLO

- **DESARROLLO**

Un analizador sintáctico, también conocido como parser, es una parte crucial de los compiladores e intérpretes en la ciencia de la computación. Su función principal es analizar la estructura gramatical de un lenguaje de programación (o cualquier otro tipo de lenguaje formal) para determinar si cumple con las reglas sintácticas definidas por el lenguaje.

Aquí están algunas de las funciones principales de un analizador sintáctico:

Análisis de la estructura gramatical: El analizador sintáctico toma el código fuente como entrada y lo divide en una estructura jerárquica que refleje la gramática del lenguaje. Esta estructura puede ser representada mediante árboles sintácticos o mediante otras estructuras de datos.

Verificación de la sintaxis: El analizador sintáctico verifica si el código fuente se ajusta a las reglas sintácticas del lenguaje de programación en cuestión. Si encuentra un error sintáctico, generalmente produce un mensaje de error indicando dónde ocurrió el error y qué tipo de error es.

Producción de un árbol sintáctico o una representación interna: A partir del código fuente, el analizador sintáctico puede construir un árbol sintáctico que represente la

estructura jerárquica del código. Este árbol es útil para las etapas posteriores del proceso de compilación o interpretación.

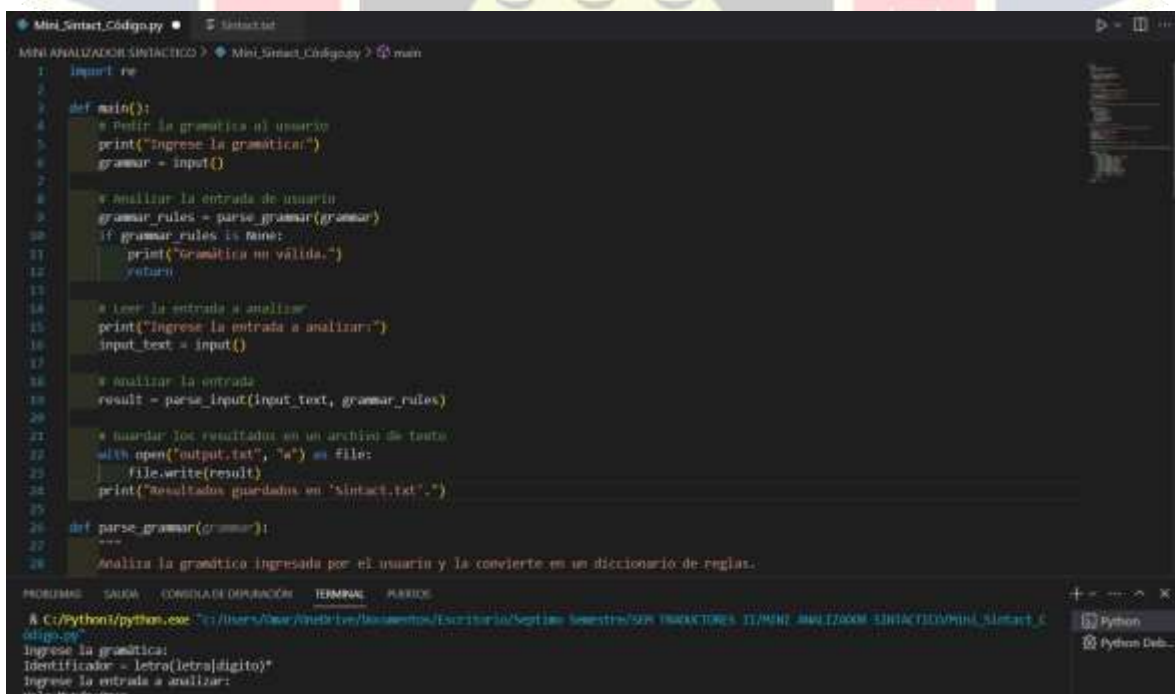
Preparación para la generación de código intermedio: En los compiladores, el analizador sintáctico es una etapa previa crucial antes de la generación de código intermedio. El árbol sintáctico o la representación interna generada por el analizador sintáctico sirve como entrada para la fase de generación de código.

Optimización sintáctica: En algunos casos, el analizador sintáctico puede aplicar optimizaciones a la estructura del código fuente para mejorar su rendimiento o su legibilidad, aunque este tipo de optimizaciones son menos comunes y suelen ser realizadas en etapas posteriores del proceso de compilación.

FUENTE:

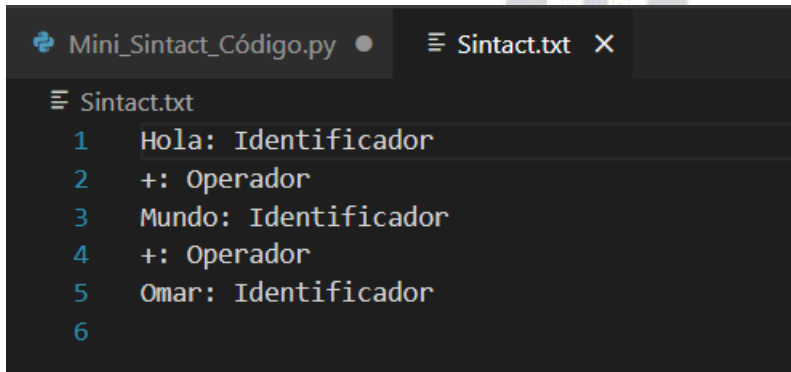
http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro32/31_funcion_del_analizador_sintctico.html

• PRUEBAS DE QUE EL PROGRAMA FUNCIONA



```
Mini_Sintact_Código.py • Mini_Sintact_Código.py
MINI ANALIZADOR SINTACTICO • Mini_Sintact_Código.py • main
1 input re
2
3 def main():
4     # pedir la gramática al usuario
5     print("ingrese la gramática:")
6     grammar = input()
7
8     # analizar la entrada de usuario
9     grammar_rules = parse_grammar(grammar)
10    if grammar_rules is None:
11        print("gramática no válida.")
12        return
13
14    # leer la entrada a analizar
15    print("ingrese la entrada a analizar:")
16    input_text = input()
17
18    # analizar la entrada
19    result = parse_input(input_text, grammar_rules)
20
21    # guardar los resultados en un archivo de texto
22    with open("output.txt", "a") as file:
23        file.write(result)
24    print("resultados guardados en 'sintact.txt'.")
25
26 def parse_grammar(grammar):
27     """
28     Analiza la gramática ingresada por el usuario y la convierte en un diccionario de reglas.
29 """
30
31 # C:\Python\python.exe "C:/Users/Usuario/Desktop/Trabajos/Segundo Semestre/SOP TRABAJOS II/MINI ANALIZADOR SINTACTICO/MINI_Sintact_Código.py"
32 Ingrese la gramática:
33 Identificador = letra(letra|digito)*
34 Ingrese la entrada a analizar:
35 HolaMundoCiao
```

Se puede ingresar diferentes Gramáticas en este programa



```
Mini_Sintact_Código.py • Sintact.txt X
Sintact.txt
1  Hola: Identificador
2  +: Operador
3  Mundo: Identificador
4  +: Operador
5  Omar: Identificador
6
```

Salida del programa (Sera guardados en un archivo Sintact.txt)

- **CONCLUSIÓN**

La función principal del analizador sintáctico es garantizar que el código fuente cumpla con las reglas sintácticas del lenguaje de programación y preparar una representación interna adecuada para las etapas posteriores del proceso de compilación o interpretación. Tuve un poco de problemas al realizar este programa ya que quise hacer interfaz, pero no he manejado ninguna, comencé a estudiar más sobre esto y me encantaría que en futuros programas emplearlos.