



Alumno: Matuz Gómez Braulio Alanni

Materia: Seminario de Traductores de lenguajes 2

sección: D02 Código estudiante: 219293637

2024A

Título de la Tarea: Mini Generador Sintáctico

como se muestra en las siguientes imágenes la implementación de lo que es el analizador sintáctico pero ahora con requerimientos extras para favorecer una auditoria diferente en la cual se toma en cuenta el cambio de los pop y push haciendo que estos usen una clase haciendo referencia a los siguientes elementos de código.

```
1  import re
2
3  class ElementoPila:
4      pass
5
6  class Terminal(ElementoPila):
7      def __init__(self, tipo, valor=None):
8          self.tipo = tipo
9          self.valor = valor
10
11  class NoTerminal(ElementoPila):
12      def __init__(self, nombre):
13          self.nombre = nombre
14
15  class Estado(ElementoPila):
16      def __init__(self, nombre):
17          self.nombre = nombre
18
19  class Alumno:
20      def __init__(self, codigo):
21          self.codigo = codigo
22
23      def muestra(self):
24          pass
25
26  class Bachillerato(Alumno):
27      def __init__(self, codigo, preparatoria):
28          super().__init__(codigo)
29          self.preparatoria = preparatoria
30
31  class Bachillerato(Alumno):
32      def __init__(self, codigo, preparatoria):
33          super().__init__(codigo)
34          self.preparatoria = preparatoria
35
36      def muestra(self):
37          print("Alumno Bachillerato")
38          print("Codigo:", self.codigo)
39          print("Preparatoria:", self.preparatoria, end="\n\n")
40
41  class Licenciatura(Alumno):
42      def __init__(self, codigo, carrera, creditos):
43          super().__init__(codigo)
44          self.carrera = carrera
45          self.creditos = creditos
46
47      def muestra(self):
48          print("Alumno Licenciatura")
49          print("Codigo:", self.codigo)
50          print("Carrera:", self.carrera)
51          print("Creditos:", self.creditos, end="\n\n")
```

```

48 class Pila:
49     def __init__(self):
50         self.lista = []
51
52     def push(self, x):
53         self.lista.insert(0, x)
54
55     def pop(self):
56         x = self.lista[0]
57         del self.lista[0]
58         return x
59
60     def top(self):
61         return self.lista[0]
62
63     def muestra(self):
64         print("Pila:", end=" ")
65         for x in reversed(self.lista):
66             x.muestra()
67         print()
68
69 # Analizador Sintáctico
70 class AnalizadorSintactico:
71     def __init__(self, tokens):
72         self.tokens = tokens
73         self.posicion = 0

```

junto con un resultado un poco inesperado debido a la mala programación  
[sintactico mod.py](#)"

```

Error sintáctico: Error sintáctico: Factor inesperado 4
PS C:\Users\USER\Downloads\Micro Analizador Lexico>

```