

UNIVERSIDAD AUTONOMA DE CHIAPAS

FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN LICENCIATURA EN INGENIERÍA EN DESARROLLO Y TECNOLOGÍAS DE SOFTWARE

MTRO LUIS ALFARO GUTIERREZ

Compiladores

Act. 1.5 Práctica . Unidad 1. Realiza un analizador Léxico en python for void burbuja arreglo

6° M

ELABORADO POR MARIA EUGENIA PEREZ PEREZ A210735

```
RUN... D No Configi > ...
                                                                                                                                                                                                                                                                                              ▷ ~ □ ..
                                                                                                                                                                                                     Ingrese su código:
                                                           import ply.lex as lex
import tkinter as tk
                                                           from tkinter import scrolledtext
from tabulate import tabulate
                                                                                                                                                                   arreglo
                                                                                                                                                                  arreglo
int
void
burbuja
static
[
]
                                                           tokens = [
| 'IDENTIFICADOR',
                                                                 'CORCHETE_DE_APERTURA',
'CORCHETE_DE_CIERRE',
                                                                                                                                                                                               Resultado del análisis léxico:
                                                                 PARENTESIS_DE_APERTURA',
PARENTESIS_DE_CIERRE',
                                                                                                                                                             'LLAVE_DE_APERTURA'
RESERVADA',
                                                                 PUNTO_Y_COMA',
'OPERADOR_DE_INCREMENTO'
                                                                                                                                                                                                                arreglo
int
void
burbuja
                                                                                                                                                                                                                static
                                                Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservado
                                                Instale la versión más reciente de PowerShell para obtener nuev
                                                PS C:\Act. 1.5 Práctica . Unidad 1. Realiza un analizador Léxico
                                                local\Programs\Python\Python\Python\Python.ex' 'c:\Users\eque\nu\\
\debugpy\adapter\..\.\\debugpy\launcher' '53022' '--' 'C:\Act.
or void burbuja arreglo\Act 1.5.py'
Carācter ilegal ';'
Carācter ilegal ';'
Carācter ilegal ';'
Carācter ilegal ';'
                                                                                                                                                                                                                                               CRLF ( Python 3.12.2 64-bit @ Go Live
```

Codigo:

```
import ply.lex as lex
import tkinter as tk
from tkinter import scrolledtext
from tabulate import tabulate
# Lista de tokens
tokens = [
    'IDENTIFICADOR',
    'TIPO_DE_DATO',
    'CORCHETE_DE_APERTURA',
    'CORCHETE_DE_CIERRE',
    'PARENTESIS_DE_APERTURA',
    'PARENTESIS_DE_CIERRE',
    'LLAVE_DE_APERTURA',
    'RESERVADA',
    'PUNTO_Y_COMA',
    'OPERADOR_DE_INCREMENTO'
```

```
# Expresiones regulares para tokens
t_IDENTIFICADOR = r'[a-zA-Z_][a-zA-Z0-9_]*'
t_CORCHETE_DE_APERTURA = r'\['
t CORCHETE DE CIERRE = r'\]'
t_PARENTESIS_DE_APERTURA = r'\('
t PARENTESIS DE CIERRE = r'\)'
t_LLAVE_DE_APERTURA = r'\{'
t PUNTO Y COMA = r';'
t OPERADOR DE INCREMENTO = r' + +
def t TIPO DE DATO(t):
    r'int'
    t.type = 'TIPO_DE_DATO'
    return t
def t RESERVADA(t):
    r'for'
    t.type = 'RESERVADA'
    return t
# Ignorar espacios y tabulaciones
t_ignore = ' \t'
# Contador de Líneas
def t newline(t):
    r'\n+'
    t.lexer.lineno += len(t.value)
# Función para manejar errores
def t error(t):
    print("Carácter ilegal '%s'" % t.value[0])
    t.lexer.skip(1)
# Construir el analizador léxico
lexer = lex.lex()
# Función para analizar el texto de entrada
def analizar(texto):
    lexer.input(texto)
    tokens encontrados = []
    while True:
        tok = lexer.token()
        if not tok:
            break # No hay más tokens
```

```
tokens encontrados.append((tok.lineno, tok.type,
tok.value))
    return tokens encontrados
# Función para manejar el evento del botón "Analizar"
def analizar texto():
    texto_entrada = entrada_texto.get("1.0", tk.END)
    tokens = analizar(texto entrada)
    texto salida.delete("1.0", tk.END)
   headers = ["Línea", "Componente léxico", "Lexema"]
    table_data = [(token[0], token[1], token[2]) for token in
tokens
    table = tabulate(table data, headers=headers, tablefmt="pipe")
    texto_salida.insert(tk.END, table)
# Crear la interfaz gráfica
ventana = tk.Tk()
ventana.title("Analizador Léxico")
frame entrada = tk.Frame(ventana)
frame entrada.pack()
etiqueta = tk.Label(frame entrada, text="Ingrese su código:")
etiqueta.pack()
entrada texto = scrolledtext.ScrolledText(frame entrada,
height=10, width=50)
entrada texto.pack()
boton analizar = tk.Button(frame entrada, text="Analizar",
command=analizar texto)
boton analizar.pack()
frame salida = tk.Frame(ventana)
frame salida.pack()
etiqueta resultado = tk.Label(frame salida, text="Resultado del
análisis léxico:")
etiqueta resultado.pack()
texto salida = scrolledtext.ScrolledText(frame salida, height=32,
width=70)
texto salida.pack()
ventana.mainloop()
```