



**UNIVERSIDAD AUTÓNOMA DE CHIAPAS**  
**FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN,**  
**CAMPUS I**

**NOMBRE DEL ALUMNO (A):**

Deycy Mercedes López Peñate.

**SEMESTRE: 6 GRUPO: "M"**

**MATRICULA: A210524**

**LICENCIATURA:**

Ingeniería en desarrollo y tecnologías de software.

**NOMBRE DEL DOCENTE:**

Dr. Luis Gutiérrez Alfaro.

**MATERIA:**

Compiladores.

**ACTIVIDAD:**

Examen.

**LUGAR:**

Tuxtla Gutiérrez, Chiapas.

**FECHA DE ENTREGA:**

13 / 02 / 2023

## CODIGO

```
import tkinter as tk
from tkinter import ttk, scrolledtext
import ply.lex as lex

class Vocabulario(object):
    palabras_reservadas = {
        'AREA': 'AREA', 'BASE': 'BASE', 'ALTURA': 'ALTURA',
    }

    tokens = [
        'IDENTIFICADOR', 'NUMERO',
        'OPERADOR', 'SIMBOLO', 'FIN',
    ] + list(palabras_reservadas.values())

    t_OPERADOR = r'[\+\*\-/\=]'
    t_SIMBOLO = r'[\(\)\[\]\{\}\;\,]'
    t_ignore = ' \t'

    def t_NUMERO(self, t):
        r'\d*\.\d+|\d+'
        try:
            t.value = float(t.value)
        except ValueError:
            print(f"Float value too large {t.value}")
            t.value = 0
        t.type = 'NUMERO'
        return t

    def t_IDENTIFICADOR(self, t):
        r'[a-zA-Z_][a-zA-Z_0-9]*'
        t.type = self.palabras_reservadas.get(t.value.upper(),
        'IDENTIFICADOR')
        return t

    def t_newline(self, t):
        r'\n+'
        t.lexer.lineno += len(t.value)

    def t_error(self, t):
        print(f"Illegal character '{t.value[0]}'")
        t.lexer.skip(1)

    def build(self, **kwargs):
        self.lexer = lex.lex(module=self, **kwargs)
```

```

def analizar_codigo():
    codigo = entrada_codigo.get("1.0", "end-1c")
    vocabulario = Vocabulario()
    vocabulario.build()
    vocabulario.lexer.input(codigo)

    for i in result_tree.get_children():
        result_tree.delete(i)

    for tok in vocabulario.lexer:

        is_reserved = "x" if tok.type in
Vocabulario.palabras_reservadas.values() else ""
        is_identificador = "x" if tok.type == 'IDENTIFICADOR' else ""
        is_numero = "x" if tok.type == 'NUMERO' else ""
        is_operador = "x" if tok.type == 'OPERADOR' else ""
        is_simbolo = "x" if tok.type == 'SIMBOLO' else ""

        # Insertar los valores en el Treeview
        result_tree.insert("", 'end', values=(tok.value, is_reserved,
is_identificador, is_numero, is_operador, is_simbolo))

ventana = tk.Tk()
ventana.title("Analizador Léxico con Tkinter")

frame_izquierda = ttk.Frame(ventana)
frame_izquierda.grid(row=0, column=0, padx=10, pady=10, sticky="nsew")

entrada_codigo = scrolledtext.ScrolledText(frame_izquierda, width=40,
height=10, wrap=tk.WORD)
entrada_codigo.pack(fill="both", expand=True)

boton_analizar = tk.Button(frame_izquierda, text="Analizar Código",
command=analizar_codigo)
boton_analizar.pack(pady=5)

frame_derecha = ttk.Frame(ventana)
frame_derecha.grid(row=0, column=1, padx=10, pady=10, sticky="nsew")

result_tree = ttk.Treeview(frame_derecha, columns=("Valor", "Palabra
Reservada", "IDENTIFICADOR", "NUMERO", "OPERADOR", "SÍMBOLO"),
show="headings")
result_tree.heading("Valor", text="Valor")

```

```

result_tree.heading("Palabra Reservada", text="Palabra Reservada")
result_tree.heading("IDENTIFICADOR", text="Identificador")
result_tree.heading("NUMERO", text="Número")
result_tree.heading("OPERADOR", text="Operador")
result_tree.heading("SÍMBOLO", text="Símbolo")

result_tree.column("Valor", anchor="center")
result_tree.column("Palabra Reservada", anchor="center")
result_tree.column("IDENTIFICADOR", anchor="center")
result_tree.column("NUMERO", anchor="center")
result_tree.column("OPERADOR", anchor="center")
result_tree.column("SÍMBOLO", anchor="center")

result_tree.pack(fill="both", expand=True)

ventana.grid_columnconfigure(0, weight=1)
ventana.grid_columnconfigure(1, weight=1)
ventana.grid_rowconfigure(0, weight=1)

ventana.mainloop()

```

## CAPTURAS

Analizador Léxico con Tkinter

area=(base*altura)/2	Valor	Palabra Reservada	Identificador	Número	Operador
Minombre Deycy fecha nacimiento 2000	area	x			
	=				x
	(				
	base	x			
	*				x
	altura	x			
	)				
	/				x
	2.0			x	
Minombre			x		
Deycy			x		
fecha			x		
nacimiento			x		
2000.0				x	