

# MANUAL TÉCNICO

## GESTOR DE CURSOS

Ciudad de Guatemala, Guatemala

**Correo electrónico:**  
ks.albizures@gmail.com

**Teléfono:** 5047-6490

## Datos de desarrollo

**Interprete utilizado:** Python 3.10.4 64-bit

**Editor de código fuente utilizado:** Visual Studio Code v1.70.2

**Sistema de gestión de paquetes “pip”**

**Interfaz gráfica usada:** Tkinter

**Método usado para la organización de widgets:** Grid (Filas y columnas)

## Objetivos

**Objetivo general:** Recibir un archivo tipo texto que siga las instrucciones de una calculadora de operaciones aritméticas y que detecte los errores léxicos posibles que pueden surgir durante su ejecución.

**Objetivos específicos:**

1. Utilizar tkinter para poder crear una interfaz dinámica y eficiente para mejorar la experiencia de usuario.
2. Leer cada instrucción de forma ordenada para la generación de operaciones solicitadas.
3. Utilizar el paradigma OOP para poder crear tokens propios con sus respectivos patrones los cuales serán utilizados para encontrar los errores léxicos del archivo.

## Descripción General:

El programa consta de la función de procesar operaciones aritméticas básicas y complejas, con el propósito de que el usuario pueda obtener la información solicitada por medio de un entorno gráfico, con la característica de mostrar también una tabla de errores léxicos y sintácticos por si el código escrito tenga algún error de estructuración.

## Paradigma utilizado

**OOP:** es un paradigma que se centra en el uso de “Objetos”, con lo cual hace que la información se contenga en forma de campos y mediante el uso de métodos se pueda gestionarlos.

Se aplicó juntos a los diccionarios, para crear funciones que permitieran la reutilización de código y realizar funciones recursivas.

### Lógica del programa:

- **Funciones de menú:** Fueron ejecutadas para cumplir las funciones de interacción de creación y manipulación del archivo txt.

```
def abrirUsuario(): ...  
  
def abrirTecnico(): ...  
  
def DatosPersonales(): ...  
  
def abrirArchivo(): ...  
  
def Guardar(): ...  
  
def GuardarComo(): ...  
  
def RutaVálida(): ...  
  
def Salir(): ...
```

- **Listado de importaciones:** La única librería que se importó fue la de tkinter debido a que se necesitaba realizar la interfaz gráfica.

```
from Ejemplo import Analizador  
from texto import Texto  
from numero import Numero # Importamos la clase Numero para la creacion del objeto Numero  
from aritmeticas import Aritmeticas # Lo mismo para la clase de creacion de Operaciones  
from operador import Operador # Los operadores Mas, Menos...  
from errores import Errores, error_L  
from estilo import Estilo, conjuntos  
from funcion import Funcion  
from Reportes import report
```

```

1  import webbrowser
2  from tkinter import filedialog
3  from tkinter import *
4  from tkinter import END
5  from tkinter import ttk
6  from tkinter import messagebox
7  from tkinter import scrolledtext
8  from tkinter import messagebox
9  from analizador_lexico import texto1

```

- **Tokens:** Los tokens utilizados para poder usarlos como palabras reservadas de nuestro programa fueron los siguientes, la letra R al inicio de cada palabra significa que son reservadas.

|    |                    |     |                                       |
|----|--------------------|-----|---------------------------------------|
| 24 | 'RTEXT0',          | 70  | t_RTIPO2 = r'TIPO'                    |
| 25 | 'RTIPO2',          | 71  | t_RTEXT02 = r'TEXT0'                  |
| 26 | 'RTEXT02',         | 72  | t_RTIPO = r'Tipo'                     |
| 27 | 'RFUNCION',        | 73  | t_RTEXT0 = r'Texto'                   |
| 28 | 'RTITULO',         | 74  | t_RFUNCION = r'Funcion'               |
| 29 | 'RDESCRIPCION',    | 75  | t_RTITULO = r'Titulo'                 |
| 30 | 'RCONTENIDO',      | 76  | t_RDESCRIPCION = r'Descripcion'       |
| 31 | 'ROPERACION',      | 77  | t_RCONTENIDO = r'Contenido'           |
| 32 | 'RCOLOR',          | 78  | t_ROPERACION = r'Operacion'           |
| 33 | 'RTAMANIO',        | 79  | t_ROPERACIONES = r'Operaciones'       |
| 34 | 'RNUMERO',         | 80  | t_RCOLOR = r'Color'                   |
| 35 | 'RSUMA',           | 81  | t_RTAMANIO = r'Tamano'                |
| 36 | 'RRESTA',          | 82  | t_RSUMA = r'SUMA'                     |
| 37 | 'RMULTIPLICACION', | 83  | t_RRESTA = r'RESTA'                   |
| 38 | 'RDIVISION',       | 84  | t_RMULTIPLICACION = r'MULTIPLICACION' |
| 39 | 'RPOTENCIA',       | 85  | t_RDIVISION = r'DIVISION'             |
| 40 | 'RMOD',            | 86  | t_RPOTENCIA = r'POTENCIA'             |
| 41 | 'RINVERSO',        | 87  | t_RMOD = r'MOD'                       |
| 42 | 'RESCRIBIR',       | 88  | t_RINVERSO = r'INVERSO'               |
| 43 | 'LLAA',            | 89  | t_RCOSENO = r'COSENO'                 |
| 44 | 'LLAC',            | 90  | t_RSENO = r'SENO'                     |
| 45 | 'IGUAL',           | 91  | t_RTANGENTE = r'TANGENTE'             |
| 46 | 'DIV',             | 92  | t_RRAIZ = r'RAIZ'                     |
| 47 | 'ENTERO',          | 93  | t_RESCRIBIR = r'ESCRIBIR'             |
| 48 | 'DECIMAL',         | 94  | t_RNUMERO = r'Numero'                 |
| 49 | 'CADENA',          | 95  | t_LLAA = r'<'                         |
| 50 | 'CORC',            | 96  | t_LLAC = r'>'                         |
| 51 | 'RAZUL',           | 97  | t_IGUAL = r'=''                       |
| 52 | 'RVERDE',          | 98  | t_DIV = r'/'                          |
| 53 | 'RROJO',           | 99  | t_CORC = r'['                         |
| 54 | 'RNEGRO',          | 100 | t_CORC = r']'                         |
| 55 | 'RAMARRILLO',      | 101 | t_RAZUL = r'AZUL'                     |
| 56 | 'RMORADO',         | 102 | t_RVERDE = r'VERDE'                   |
| 57 | 'RNARANJA',        | 103 | t_RROJO = r'ROJO'                     |
| 58 | 'RANARANJADO',     | 104 | t_RNEGRO = r'NEGRO'                   |
| 59 | 'RCOSENO',         | 105 | t_RAMARRILLO = r'AMARRILLO'           |
| 60 | 'RSENO',           | 106 | t_RMORADO = r'MORADO'                 |
| 61 | 'RTANGENTE',       | 107 | t_RNARANJA = r'NARANJA'               |
| 62 |                    | 108 | t_RANARANJADO = r'ANARANJADO'         |
|    |                    | 109 |                                       |

- **GRAMATICA:** Esta basado en instrucciones que almacenan por dentro un conjunto de tokens, algunas instrucciones se llaman así mismas para poder ejecutarlas dentro de sí, esto se usa sobre todo en las operaciones. Fragmento:

```

178 def p_init(t):
179     'init' : instrucciones'
180     t[0] = t[1]
181     return t[0]
182
183 def p_instrucciones_lista(t):
184     '''instrucciones : instrucciones instruccion
185     | | | | | instruccion'''
186     if len(t) == 2:
187         t[0] = [t[1]]
188     else:
189         t[1].append(t[2])
190         t[0] = t[1]
191
192 def p_instruccion(t):
193     '''instruccion : INSTIPO
194     | | | | | INSTEXTO
195     | | | | | INSTFUNCION
196     | | | | | INSTESTILO
197
198
199     '''
200     t[0] = t[1]
201
202 def p_instruccionTipo(t):
203     'INSTIPO : LLAA RTIPO LLAC instrucciones_2 LLAA DIV RTIPO LLAC'
204     t[0] = t[4]
205
206 def p_instruccionTexto(t):
207     #global textol
208     'INSTEXTO : LLAA RTEXTO LLAC CADENA LLAA DIV RTEXTO LLAC'
209     t[0] = Texto(t[4], t.lineno(1), find_column(input,t.slice(1)))
210     #textol=t[4]
211     print(t[4])
212
213 def p_instruccionFuncion(t):
214     'INSTFUNCION : LLAA RFUNCION IGUAL RESCRIBIR LLAC instrucciones_2 LLAA DIV RFUNCION LLAC'
215     t[0] = Funcion(t[6][0], t[6][1], t[6][2], t.lineno(1), find_column(input,t.slice(1)))
216

```

- **Reportes:** los reportes creados se generaron con la ayuda de algunas listas las cuales almacenaron los errores, resultados y estilos para poder modificar el archivo html.

```

6 global error_L
7
8 texto=""<html>\n"
9
10 texto=""<title> Reporte Error </title>\n"
11
12 texto=""<font size=2>\n"
13
14 texto=""<font face="Times New Roman">\n"
15
16 texto=""<font color=orange>\n"
17
18 texto=""<body style="background-color:#002045">\n"
19
20 texto=""<div align=center> REPORTE DE ERRORES </div>\n"
21
22 texto=""</font>\n"
23
24 texto=""</font>\n"
25
26 texto=""<font color=white>\n"
27
28 texto=""<div align=center> </div>\n"
29
30 texto=""</font>\n"
31
32 texto=""<font size=2>\n"
33
34 texto=""<table border=1 align=center TR BGCOLOR="#1A5760" TABLE BORDERCOLOR="Black">\n"
35
36 texto=""<thead>\n"
37
38 texto=""<tr>\n"
39
40 texto=""<thNo.</th>\n"
41
42 texto=""<thLexema</th>\n"
43
44 texto=""<thTipo</th>\n"
45
46 texto=""<thFila</th>\n"
47
48 texto=""<thColumna</th>\n"
49
50 texto=""</tr>\n"
51
52 texto=""</thead>\n"
53
54 texto=""<tbody>\n"
55
56 if len(error_L)==0:
57     texto=""<font color=white>\n"
58     texto=""<div align=center> NO SE ENCONTRARON ERRORES B </div>\n"
59     texto=""</font>\n"
60
61 else:
62     texto=""<font color=white>\n"
63     texto=""<div align=center> OH NO, SE ENCONTRARON ERRORES B( </div>\n"
64     texto=""</font>\n"
65     for j in range(0,len(error_L)):
66         texto=""<tr>\n"
67         texto=""<td>\n"+str(j+1)+"</td>\n"
68         texto=""<td>\n"+str(error_L[j]]["Lexema"])+"</td>\n"
69

```



## AUTOMATA FINITO DEL ANALIZADOR LEXICO

