

**UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**  
**FACULTAD DE INGENIERIA**  
**LENGUAJES FORMALES Y DE PROGRAMACION**  
**LABORATORIO**



**PROYECTO NO.1**  
**MANUAL TECNICO**

**Jonathan Josue Alvarez Chacon**

**Carnet: 202100259**

# 1. Interfaz Grafica

Para la realización de la interfaz grafica se utilizo la librería que trae python **TKINTER**, la cual nos ofrece muchas herramientas y nos facilitara el programado de la ventana interactiva.

```
ventana2 = Tk()
ventana2.title("COMPILADOR LEXICO EPICO")
ventana2.geometry("800x600")
EtiquetaArchivo = Label(text="ARCHIVO", font=("Consolas 16"))
EtiquetaAyuda = Label(text="AYUDA", font=("Consolas 16"))
EtiquetaCarga = Label(text="Archivo Cargado con Exito!", font=("Consolas 14"), fg="green")
EtiquetaTemasAyuda = Label(text="Jonathan Josue Alvarez Chacon", font=("Consolas 14"))
EtiquetaTemasAyuda2 = Label(text="Facultad de Ingenieria, USAC", font=("Consolas 14"))
EtiquetaGuardar = Label(text="Documento Guardado!", font=("Consolas 14"), fg="green")
Caja = Text(ventana2,width=50,height=20, bd=4, selectbackground="red")
```

En este bloque el objeto “ventana2” es nuestra ventana y con las distintitas opciones nos encargamos del titulo y del tamaño de la misma, y a continuación se establecen unas etiquetas que desplegaremos luego y nuestro objeto “Caja” que es el bloque de texto.

```
#DECLARANDO LOS BOTONES-----
botonLimpiar = Button(ventana2, text="Limpiar", width=20, height=2, bg="white", command=Limpiar)
botonGuardar = Button(ventana2, text="Guardar", width=20, height=2, bg="lightgreen", command=Guardar)
botonAbrir = Button(ventana2, text="Abrir", width=20, height=2, bg="purple", foreground="white", command=Abrir)
botonGuardarComo = Button(ventana2, text="Guardar Como", width=20, height=2, bg="lightgreen", command=GuardarComo)
botonAnalizar = Button(ventana2, text="Analizar", width=20, height=5, bg="lightblue", command=Analizar)
botonErrores = Button(ventana2, text="Errores", width=20, height=2, bg="Red", foreground="white")
botonSalir = Button(ventana2, text="Salir", width=20, height=2, bg="Orange", command=Salir)
botonManualU = Button(ventana2, text="Manual de Usuario", width=20, height=2, bg="darkgray", command=ManualUsuario)
botonManualT = Button(ventana2, text="Manual Tecnico", width=20, height=2, bg="darkgray", command=ManualTecnico)
botonTemas = Button(ventana2, text="Temas de Ayuda", width=20, height=2, bg="darkgray", command=TemasAyuda)
#CajadeTexto = Entry(ventana2, font=("Consolas", 20), width=55, selectforeground="white", selectbackground="red")
```

Se declaran los botones para colocarlos en nuestra ventana

```
#COLOCANDO LOS BOTONES EN LA VENTANA-----  
#EtiquetaAyuda.grid(row=2, column=0)  
botonAbrir.grid(row=1, column=1, padx=10, pady=10)  
botonGuardar.grid(row=2, column=1, padx=10, pady=10)  
botonGuardarComo.grid(row=3, column=1, padx=10, pady=10)  
botonAnalizar.grid(row=0, column=1, padx=10, pady=10)  
botonLimpiar.grid(row=0, column=2, padx=10, pady=10)  
botonErrores.grid(row=4, column=1, padx=10, pady=10)  
botonSalir.grid(row=4, column=2, padx=10, pady=10)  
botonManualU.grid(row=1, column=2, padx=10, pady=10)  
botonManualT.grid(row=2, column=2, padx=10, pady=10)  
botonTemas.grid(row=3, column=2, padx=10, pady=10)  
#CajadeTexto.grid(row=1, column=3, padx=10, pady=10)  
Caja.grid(row=0, column=0, padx=10, pady=10)  
  
ventana2.mainloop()  
  
#RE COLOCAR TODOS LOS BOTONES Y LAS WEAS PARA QUE SE MIRE CHIDO
```

Con la función `.grid` convertimos la ventana en una de posiciones y gracias a las funciones predeterminadas de `row`, `column` podemos orientar los objetos donde sea, la última función es importante para establecer el loop de la ventana que impida cerrarse.

## 2. Botones

Gracias a que TKINTER nos permite trabajar con funciones para añadirlas como comandos en los botones se utilizó una función para cada comportamiento de los botones:

```
def TemasAyuda():  
    EtiquetaTemasAyuda.grid(row=1, column=0)  
    EtiquetaTemasAyuda2.grid(row=2, column=0)  
  
def ManualTecnico():  
    path = "Manual_Tecnico.pdf"  
    os.system(path)  
  
def ManualUsuario():  
    path2 = "Manual_De_Usuario.pdf"  
    os.system(path2)  
  
def Guardar():  
    print("GUARDADO...")  
    entrada = Caja.get("1.0","end")  
    archivo2 = open(filename, "w")  
    archivo2.write(entrada)  
    archivo2.close()  
    r = mb.askokcancel("Mensaje", "Archivo guardado exitosamente")  
    #EtiquetaGuardar.grid(row=3, column=0)
```

Estos son unos ejemplos de algunos botones, se estableció un “def” para cada uno y se comenzó a codificar las instrucciones de cada uno.

### 3. Autómata Finito Determinista

Para la creación de nuestro analizador léxico se optó por utilizar un autómata que determinara y almacenara cada token de nuestro documento de texto para posteriormente ser analizado y ejecutado, para esta sentencia se utilizó una gran cadena de “if” y “elif” que determinaran cada caso posible y poder almacenar la información de manera correcta en unos arreglos y variables agregados.

```
while len(cadena) > 0:
    char = cadena[0]

    #ignorando espacios en blanco
    if char == "\n":
        self.fila += 1
        self.columna = 0
        cadena = cadena[1:]
        continue
    elif char == " ":
        self.columna += 1
        cadena = cadena[1:]
        continue

    if self.estado_actual == 0:
        if char == "{":
            self.guardar_token(char)
            self.estado_anterior = 0
            self.estado_actual = 1

    elif self.estado_actual == 1:
        if char == "'":
            self.guardar_token(char)
            self.estado_anterior = 1
            self.estado_actual = 2
        elif char == "{":
            self.guardar_token(char)
            self.estado_anterior = 1
            self.estado_actual = 10
```

Se creo una clase para el autómata aparte para facilitar su redacción y poder trabajar con mayor comonidad, este es un ejemplo de los primeros casos que analiza nuestro programa, se trabajo por tipo de casos y el código es extenso por lo que solo se muestra un poco de su interior en este documento.



### 3.1 Lexemas Analizados:

Una vez se analice el archivo se almacenaran los datos en una variable la cual imprimiremos en consola con la siguiente función, esta nos permite visualizar el orden que los tokens fueron almacenados

```
def imprimir_lexemas(self):  
    for token in self.tabla_tokens:  
        print(token.lexema)  
    self.tabla_tokens = []  
    self.cadena = ""  
    self.fila = 0  
    self.columna = 0  
    self.estado_actual = 0  
    self.estado_anterior = 0  
    self.estados_aceptacion = [9]  
    self.token = ""
```

### 3.2 Detectar Operación:

Posteriormente a los resultados se verifica si existen algunas operaciones en nuestro documento por lo que la siguiente función detecta si las palabras clave de las operaciones se encuentra dentro de nuestro documento y por lo tanto lo imprime en consola.

```
def detectar_operacion(self):  
    for token in self.tabla_tokens:  
        if token.lexema == "Suma":  
            print("se encontro una suma")  
        elif token.lexema == "Resta":  
            print("se encontro una resta")  
        elif token.lexema == "Multiplicacion":  
            print("se encontro una multiplicacion")  
        elif token.lexema == "Division":  
            print("se encontro una division")  
        elif token.lexema == "Potencia":  
            print("se encontro una potencia")  
        elif token.lexema == "Raiz":  
            print("se encontro una raiz")  
        elif token.lexema == "Seno":  
            print("se encontro un seno")  
        elif token.lexema == "Coseno":  
            print("se encontro un coseno")  
        elif token.lexema == "Tangente":  
            print("se encontro una tangente")
```