

# MANUAL TÉCNICO

Universidad San Carlos de Guatemala  
Lenguajes Formales y Programación B-  
Pablo Josué Barahona Luncey  
202109715  
3560855890101

PROYECTO 1



# Acercas del programa

---

Este programa fue desarrollado con el lenguaje formal de programación Python, el objetivo del programa fue crear un analizador léxico por medio de tókens.

En este programa se usaron distintas librerías y se realizó con diferentes métodos

La librería principal fue Tkinter, debido a que se tenía que hacer interfaz gráfica

# Métodos

---

Para el menú principal utilizamos la librería Tkinter y también para toda las funcionalidades gráficas del programa.

El programa posee dos archivos, con nombre "Operadora" que es la clase main y "analizador"

## Clase Operadora

- `abrirArchivo()`

Se hizo un método para abrir archivo, el cuál se inicia al presionar el botón "Abrir Archivo", se utilizó el importe `filedialog` y `nthpath`

- `guardar()`

Este método guarda el archivo con el mismo nombre del original.

- `guardarcomo()`

Este método guarda el archivo con un nuevo nombre

- `analizar()`

Este método llama a la clase Analizar, la cual se especifica más adelante y también abre el reporte PDF de Analizar

- `errores()`

Este método llama a otro método de la clase Analizar y también abre un archivo PDF generado en la otra clase

- `salirPro()`

Método que sale del programa

- `manualusuario()`

Este método abre el manual de usuario

- `manualtecnico()`

Este método abre el manual técnico

- `info()`

Este método muestra la información del programador

# analizador.py

---

Este archivo maneja 2 clases llamadas L\_tokens y Analizador

## Clase L\_tokens

Esta clase almacena los tokens para el autómata, acá utilizamos la librería "enum" y su función es especificar los valores que se esperan al leer el archivo de entrada

## Clase Analizador

Este es el núcleo del programa.

- `__init__()`

Este método inicia las variables a utilizar

- `quitar()`

Este método quita las líneas leídas

- `aumentarLinea()`

Este método lee las líneas siguientes

- `Numero()`

Este método interpreta los tokens por orden, el método es para la etiqueta Números, en este método se evalúan los números con la operación obtenida en el método Operacion, (ej: SUMA de 1 y 2)

- `Operacion()`

Este método interpreta los tokens por orden, el método es para la etiqueta Operacion, en este método se evalúan las operacionales (ej: etiqueta SUMA)

- `Texto()`

Este método interpreta los tokens por orden, el método es para la etiqueta Texto, acá se analiza el texto que se otorga en el archivo de entrada

# analizador.py

---

- Funcion()

Este método interpreta los tokens por orden, el método es para la etiqueta Funcion, se evalúan el orden que se va a generar el HTML y su título respectivo

- Estilo()

Este método interpreta los tokens por orden, el método es para la etiqueta Estilo, muestra el tamaño y color de letra del HTML

- Tipo()

Este método interpreta los tokens por orden, el método es para la etiqueta Tipo

- compile()

Este método inicia el programa y la clase analizador, empieza leyendo el texto y hacerle arreglos respectivos

- htmlanalizar()

Este HTML es el encargado de generar el HTML de los resultados operativos que se obtienen

- htmlerrores()

Este método se encarga de generar el HTML de los errores generados

# AFD

Esta es una muestra gráfica del AFD con el orden de nuestro programa y el uso de los tokens

