

MANUAL TÉCNICO

**PROYECTO 1: LABORATORIO DE
LENGUAJES FORMALES Y DE
PROGRAMACIÓN**

MANUAL TÉCNICO

Un manual técnico es un documento que va dirigido a un público con conocimientos técnicos sobre el área en la que estamos trabajando, en este caso, hablando de un proyecto de programación, va dirigido a una persona que tenga conocimientos en programación.

Este documento debe contener toda la información sobre los recursos utilizados en el proyecto, el equipo utilizado, y una explicación detallada de todo lo que se ha realizado dentro del proyecto.

Normalmente un manual técnico va dirigido a otros programadores o administradores del sistema, con la finalidad de que se pueda conocer de mejor manera cómo se realizó el programa y cómo funciona, para poder darle mantenimiento cuando se requiera, o hacer cambios, sin que la persona que desarrolló el sistema esté presente.

LUGAR, FECHA Y RESPONSABLES DE LA ELABORACIÓN

Esta aplicación se realizó en la ciudad de Guatemala departamento de Guatemala, la fecha de elaboración inicio el 10 de marzo de 2023, culminando el 27 de marzo de 2023, la aplicación fue desarrollada por el estudiante de la Facultad de Ingeniera de la Universidad de San Carlos de Guatemala David Eduardo Matias Ramirez.

OBJETIVOS Y ALCANCES DEL SISTEMA

Objetivos Generales:

- Implementar por medio de estados un analizador léxico.
- Utilizar funciones de manejo de cadenas de caracteres en lenguaje Python.
- Programar un Scanner para el análisis léxico.
- Construir un scanner basándose en un autómata finito determinístico.
- Crear una herramienta para interactuar de forma visual con el usuario con Tkinter

La aplicación va dirigida para todas las personas que por medio de un archivo de texto con extensión txt, json, etc; que contenga información sobre diversas operaciones aritméticas y trigonométricas simples o complejas quieran poder conocer los resultados de estas operaciones puedan hacerlo de una manera clara y amigable con ellos; al hacer uso de la aplicación desarrollada se podrá cumplir con sus expectativas.

Como objetivo principal se tiene el poder desarrollar un software que implemente un analizador léxico para la lectura de los archivos de entrada y que pueda ir gestionando las operaciones a ejecutar y los errores de caracteres no reconocidos. Facilitando la forma visual de la gestión del archivo como la salida de los resultados y errores.

ESPECIFICACION TÉCNICA

- 512 mb de espacio en disco duro.
- 512 mb de memoria ram.
- Windows 7 o versiones en adelante.
- Un IDE de programación.
- Equipo Intel Pentium o superior.

PARADIGMAS DE PROGRAMACION UTILIZADOS

La versión de Python utilizada para la elaboración de esta práctica es la vr 3.9.1

Realizada en Windows 10 x64. Disco duro de 500GB. Memoria RAM de 4GB.

A continuación se presentan las clases, métodos utilizados y elaborados en la practica 1, así como una breve descripción de ellos.

Variables Utilizadas:

Lista Lexemas[]; Nos ayudo a guardar los lexemas que se fueron construyendo.

Lista Errores[]; Nos ayudo a guardar los errores que se fueron generando.

Puntero; Nos ayudo a ir leyendo la posición de la cadena.

Caja Texto; Contuvo el texto del archivo de entrada.

Filename; Contuvo el archivo cargado en el sistema.

N_linea; Contuvo el numero línea en que se posicionaba el puntero.

N_Columna; Contuvo el numero línea en que se posicionaba el puntero.

Paquetes:

Se hizo uso de 3 paquetes, los cuales llevan de nombre Abstract, Instrucciones y Errores.

Clases:

Clase Pantalla Principal: Es la clase general de alguna manera, ya que en ella se concentra toda la interfaz gráfica de la aplicación y en ella se definen los métodos que nos ayudan en las demás funcionalidades del sistema.

Clase Expression: Almacenada en (Abstract) Esta clase nos sirve para poder ir creando las demás, ya que es una clase abstracta y esto nos ayuda en el uso de la recursividad.

Clase Lexema: Almacenada en (Abstract) Esta clase nos ayuda para poder darle los atributos necesarios que tendrán nuestros lexemas cuando se vayan creando.

Clase Numero: Almacenada en (Abstract) Esta clase nos ayuda para poder darle los atributos necesarios que tendrán nuestros números cuando se vayan leyendo y guardando en el analizador.

Clase Error: Almacenada en (Errores) Esta clase nos ayuda para poder darle los atributos necesarios que tendrán nuestros errores cuando se vayan generando si es que existieran.

Clase Aritméticas: Almacenada en (Instrucciones) Esta clase sirve para poder realizar las operaciones aritméticas simples o complejas según las contenga el archivo de entrada.

Clase Trigonométrica: Almacenada en (Instrucciones) Esta clase sirve para poder realizar las operaciones trigonométricas según las contenga el archivo de entrada.

Main:

Interfaz: En este archivo se desarrollo toda la interfaz grafica del programa, por medio de la clase Pantalla Principal se creo una venta de nombre PP por medio de TK() que proporciona la librería de

Tkinter. En Pantalla Principal se manda a llamar al método pantalla_1 la cual contiene todos los componentes que forman parte de la interfaz grafica como lo son los botones de Abrir, Guardar, Guardar Como, Analizar, Errores, Salir, Manual de Usuario, Manual Técnico y Temas de Ayuda.

Botones de Pantalla 1:

Abrir: Al presionar este botón se nos permite abrir el explorador de archivos y seleccionar el archivo que vamos a analizar en el programa. Esto sucede gracias al método abrir_archivo que utiliza el modulo askopenfilename de Python, también almacena el contenido del archivo en una variable llamada texto y luego inserta el contenido de esta variable en una Caja de Texto que se encuentra en Pantalla 1.

Guardar: Al presionar este botón se nos permite guardar cualquier cambio realizado al archivo de entrada en la caja de texto para poder analizarlo o guardar como. Esto es posible por medio de los métodos get, write y close de un archivo.

Guardar Como: Al presionar este botón se nos permite abrir el explorador de archivo y cambiar el nombre y la extensión del archivo que estamos analizando y nos genera uno nuevo con el nombre y la extensión seleccionados.

Analizar: Al presionar este botón se nos genera una imagen svg con las operaciones realizadas en orden jerárquico y con el resultado final correspondiente. Este botón recibe un método llamado ejecutar el cual contiene el método graficar que es el encargado de generar la imagen svg por medio de lo recibido del método instrucción el cual esta definido en el archivo analizador_lexico. El método instrucción recibe varios métodos mas como armar_numero, armar_lexema entre otros los cuales hacen posible la lectura del archivo carácter

por carácter y número por número hasta poder realizar las operaciones y generar la gráfica.

Errores: Al presionar este botón se nos genera un archivo JSON con los caracteres que el programa no reconoce y estos los toma como errores.

Manual De Usuario: Nos muestra el manual de usuario en el navegador por medio del método `webbrowser.open_new` importado de `webbrowser`.

Manual Técnico: Nos muestra el manual técnico en el navegador por medio del método `webbrowser.open_new` importado de `webbrowser`.

Temas de Ayuda: Muestra información del estudiante desarrollador por medio del método `pantalla_información` la cual genera una nueva ventana con etiquetas y un botón que ayuda a regresar a la pantalla principal.

Salir: Cierra la aplicación destruyendo la ventana que contiene a Pantalla principal por medio del método `destroy()`