

Universidad Rafael Landívar

Facultad de Ingeniería

Lenguajes Formales y Autómatas, Sección 02

Docente: Mgtr. Vivian Damaris Socorro Campos Gonzales

Manual Técnico

“Analizador de Operaciones Aritméticas”

Versión 3.5

Autores:

Lizbeth Andrea Herrera Ortega – 1246024

Marcela Nicole Letran Lee – 1102124

Guatemala, 02 de noviembre de 2025

Requisitos para la Compilación y Ejecución

Requisitos de Hardware

El sistema está diseñado para ejecutarse en entornos de escritorio o portátiles con capacidades básicas de procesamiento.

Componente	Requisitos mínimos	Requisitos recomendados
CPU	Procesador de 2 núcleos (x86_64)	Intel i5 o superior
RAM	2 GB	4 GB o más
Almacenamiento	100 MB libres	SSD con 500 MB disponibles
Periféricos	Teclado, pantalla y mouse para navegación	Teclado, pantalla y mouse para navegación

Requisitos de Software

Sistema desarrollado en Python, con bibliotecas estándar propias del mismo. Compatible con:

- Windows 10 o superior
- MacOS 11 (Big Sur) o superior
- Distribuciones Linux (Ubuntu 20.04+, Debian)

Necesario contar con un interprete de Python, con una versión mínima de 3.10.

No se requiere alguna instalación adicional, ya que se utilizan bibliotecas estándar del lenguaje:

- os: Permite el manejo de rutas y archivos en el sistema operativo.
- datetime: Para generación de fechas de creación al momento de ejecución.
- tkinter: Para la creación de la interfaz grafica (GUI).
- filedialog: Abrir cuadro de dialogo para seleccionar archivos.
- messagebox: Para mostrar mensajes emergentes al usuario.
- scrolledtext: Incluye el área de texto con barra de desplazamiento, para la edición del archivo a analizar.
- webbrowser: Para abrir los reportes en HTML generados en el navegador predeterminado del sistema.

Editor recomendado:

Visual Studio Code, PyCharm o cualquier editor con soporte para Python 3.10+.

Funcionamiento interno

El programa tiene como objetivo analizar y ejecutar instrucciones aritméticas contenidas en un archivo de código fuente, detectando posibles errores léxicos y generando diferentes archivos de salida que incluyen los resultados de las operaciones, los errores detectados y un diagrama jerárquico de las expresiones aritméticas. Además, incluye una interfaz gráfica intuitiva que facilita la interacción del usuario y la visualización de los resultados.

El sistema está estructurado en tres componentes principales:

1. Interfaz Gráfica de Usuario (GUI)

Está diseñada para ser simple, clara y funcional, con botones de control dispuestos en la parte superior y lateral de la ventana principal.

Botones superiores

- Abrir Archivo: Permite seleccionar y cargar un archivo de código fuente desde el explorador de archivos.
- Guardar Archivo: Guarda el contenido del área de texto sobre el archivo actual.
- Guardar Como: Guarda el código fuente en un nuevo archivo con nombre diferente.
- Analizar Archivo: Envía el contenido del área de texto al módulo de análisis léxico y sintáctico para la ejecución de operaciones.

Panel Lateral Derecho

Documentación:

- Manual de Usuario: Abre un archivo PDF con la guía básica de uso del programa.
- Manual Técnico: Muestra la documentación técnica del sistema, en formato PDF.

Resultados finales:

- Resumen General: Abre un archivo HTML con un resumen de la cantidad de resultados que incluye: operaciones válidas, errores léxicos y gráficos por operación.
- Resultados Operaciones: Abre un archivo con los resultados de las operaciones aritméticas del archivo analizado, junto al árbol jerárquico (SVG), en formato HTML.
- Errores Léxicos: Abre un archivo con los errores léxicos compilados durante el análisis, en formato HTML.

Ayuda:

- Proporciona información sobre los desarrolladores de la aplicación.

Área de texto

Espacio central donde el usuario puede escribir, visualizar o editar el código fuente antes de procesarlo. El contenido de esta área es el que se analiza y ejecuta.

Cuando el usuario presiona el botón Analizar Archivo, el contenido del área de texto se almacena temporalmente en un archivo de entrada, el cual se envía al módulo de análisis. Los resultados del proceso (operaciones ejecutadas, errores léxicos y diagrama de operaciones) son generados automáticamente y pueden visualizarse desde los archivos de salida correspondientes.

2. Análisis léxico y sintáctico para ejecución de instrucciones

Recibe el contenido del área de texto en forma archivo y realiza las siguientes operaciones:

1. Análisis Léxico

Se identifican los tokens válidos en el código fuente, detectando errores léxicos.

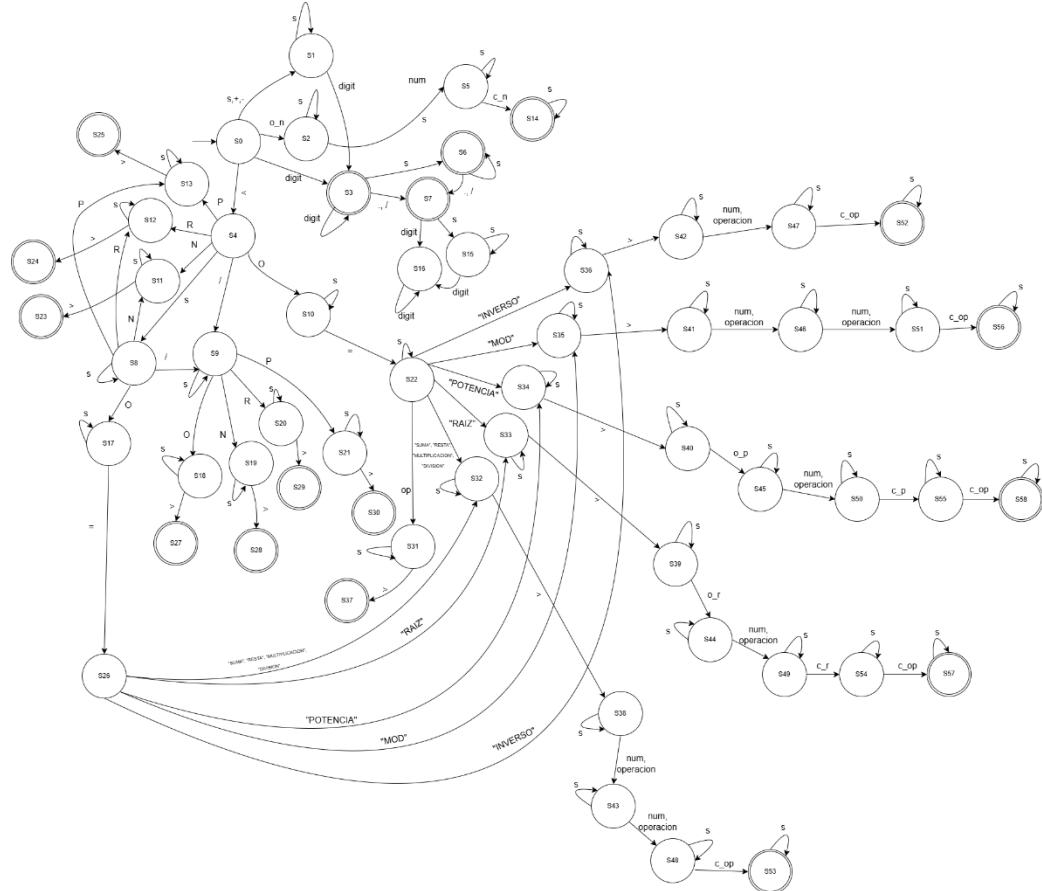
Tokens aceptados

Se desarrollaron los tokens válidos para la ejecución del programa, en base a las operaciones principales de: suma, resta, multiplicación, división, potencia, raíz, mod, inverso.

Token	Descripción
open_operacion	Indica el inicio de una operación.
close_operacion	Indica el cierre de una operación.
open_potencia	Indica el inicio de una operación de potencia.
close_potencia	Indica el cierre de una operación de potencia.
open_raiz	Indica el inicio de una operación de raíz.
close_raiz	Indica el cierre de una operación de raíz.
open_numero	Marca el inicio de un número.
close_numero	Marca el final de un número.
valor	Representa cualquier valor numérico entero o decimal.
Numero	Incluye la expresión completa para representar un número a operar (apertura, valor, cierre).
SUMA, RESTA, MULTIPLICACION, DIVISION	Operaciones aritméticas básicas.
POTENCIA	Operación de potencia.
RAIZ	Operación de raíz.
MOD	Operación de modulo.
INVERSO	Operación de inverso.

Autómata Finito Determinista

Por medio del siguiente AFD, se reconocen los elementos básicos del lenguaje (tokens aceptados).



Cada símbolo de entrada se procesa según la tabla de transiciones (Adjunta en archivo “PRIMERA FASE/Tabla de Follow y Transiciones.xlsx”), que define los cambios de estado en función del carácter leído y el estado actual.

Reconocimiento de tokens

Desde el código fuente, se recorren las cadenas de entrada carácter por carácter. Se genera un token válido al alcanzar un estado de aceptación. Si no existe una transición válida desde el estado actual, se genera un error léxico que se almacena con su posición en fila y columna.

2. Análisis sintáctico

Procesa la lista de tokens generados, evaluando las operaciones aritméticas. Respeta la jerarquía de operadores, y construye un árbol sintáctico que representa la estructura de las expresiones matemáticas.

3. Generación de reportes

Se generan los archivos de salida luego del proceso de análisis.

- Se genera un archivo HTML con un resumen general de operaciones ejecutadas, errores léxicos almacenados y gráficos por operación ejecutada.
- Se genera un archivo HTML con los resultados numéricos de las operaciones validadas, creando un gráfico (SVG) con el árbol jerárquico de cada resultado.
- Se exporta un archivo HTML con todos los errores léxicos detectados.

Diagrama de Flujo de Funcionamiento Interno

