

Universidad San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias y Sistemas
Lenguajes Formales y de Programación
Ing. Daniel Gonzales



MANUAL DE USUARIO

PROYECTO #1

Henry Josué Argueta Champet
Reg. academico: 202131261
Fecha: 02/10/2025

ÍNDICE

| | |
|--|---|
| Descripción de la actividad..... | 3 |
| En qué consiste el la aplicación..... | 3 |
| Interfaz Principal del Editor..... | 4 |
| Carga de Archivo de Entrada..... | 5 |
| Cómo Funciona el Analizador: El Proceso Interno..... | 6 |
| 1. Preparación y Escaneo (El Motor)..... | 6 |
| 2. Reconocimiento de Patrones (El Autómata)..... | 7 |
| 2. Reconocimiento de Patrones (El Autómata)..... | 7 |
| 3. Recuperación de Errores..... | 8 |

Descripción de la actividad.

Una de las tareas de un analizador léxico es la detección y reconocimiento de tokens válidos para el lenguaje en cuestión, así como el reconocimiento de errores de tokens no definidos o caracteres que no se encuentran en el alfabeto del lenguaje. El analizador léxico también debe reaccionar correctamente ante los posibles errores que se encuentren, indicando la posición del error y una descripción del mismo.

La siguiente práctica invita al estudiante a idear algoritmos por medio de un autómata que permitan reconocer patrones en una cadena de caracteres y reportar errores que se presenten a lo largo del análisis.

En qué consiste el la aplicación.

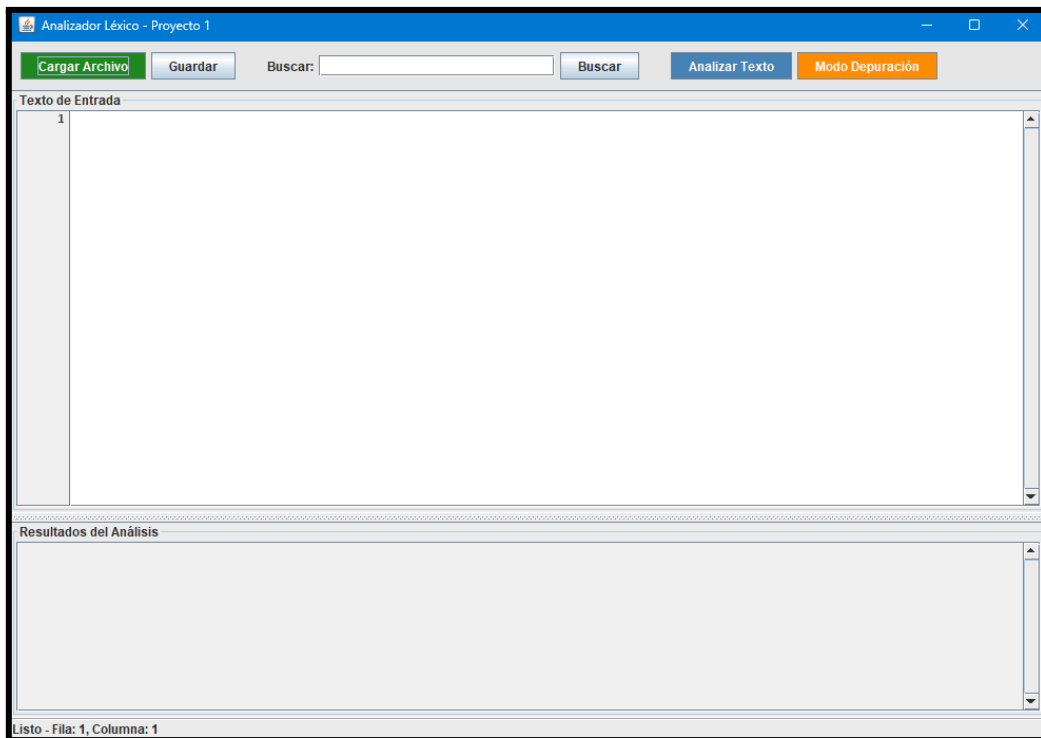
Esta aplicación es un Analizador Léxico, diseñado para procesar código fuente escrito en un lenguaje específico. Mi objetivo principal es la detección y reconocimiento de tokens válidos para nuestro lenguaje, así como la identificación precisa de cualquier error léxico o carácter no definido en nuestro alfabeto.

La herramienta provee una interfaz gráfica completa que no solo permite escribir y analizar código, sino también gestionar archivos, buscar patrones, y utilizar un potente Modo Depuración para observar el proceso de análisis paso a paso.

Interfaz Principal del Editor

Al iniciar la aplicación, se presenta la ventana principal (LexerGUI), la cual se divide en dos áreas principales y una barra de controles:

- Área de Texto de Entrada (textArea): Es la zona principal donde el usuario puede escribir directamente su código fuente o cargar un archivo para su análisis. Este componente está diseñado para mostrar en todo momento la fila y columna actual del cursor, facilitando la ubicación de errores.
- Área de Resultados y Reportes (resultArea): Aquí se mostrará la información generada tras ejecutar el análisis, incluyendo la lista de tokens, el reporte de errores o el recuento de lexemas.
- Controles y Botones: Una barra superior o lateral contiene los botones para ejecutar las funciones clave.



Carga de Archivo de Entrada

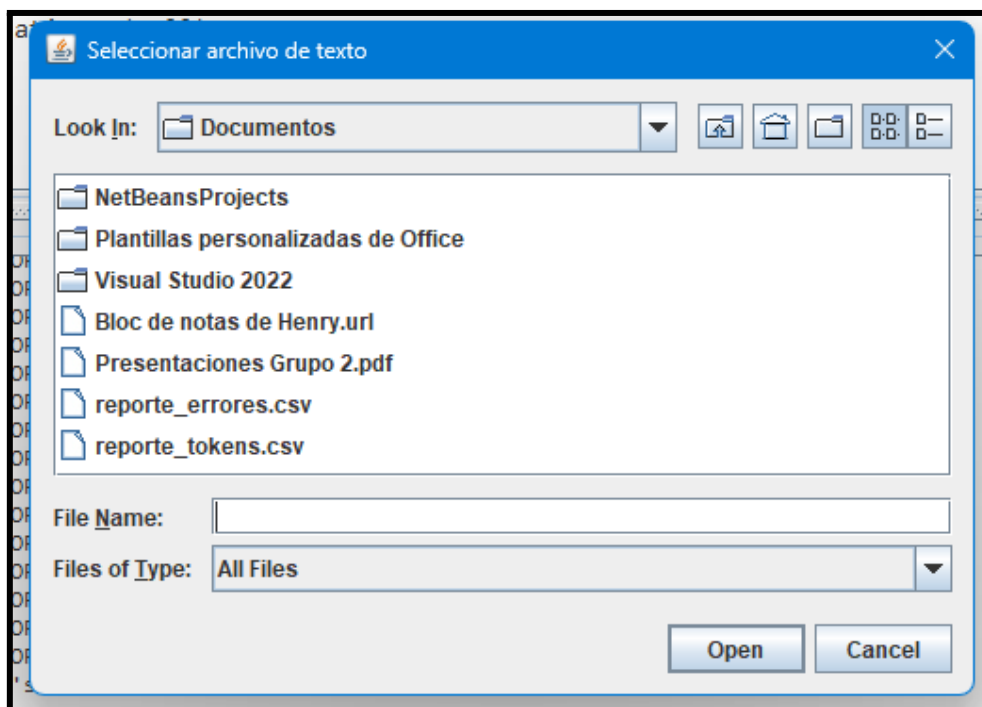
Para facilitar la edición y análisis de código extenso, nuestro analizador permite importar el texto directamente desde un archivo.

Al presionar el botón "Cargar Archivo" en la barra superior, la aplicación abre una ventana estándar de selección de archivos del sistema operativo.

Flujo de la Función:

- Selección: Aparece una ventana de "Seleccionar archivo de texto" que le permite navegar por sus carpetas y elegir el archivo de código fuente deseado.
- Importación: Una vez que selecciona el archivo y hace clic en "Abrir", el contenido completo de ese archivo de texto se carga automáticamente en el Área de Texto de Entrada de la aplicación, quedando disponible para su edición, análisis y coloreado automático.

Esta característica garantiza que pueda trabajar con archivos externos de manera fluida y sin la necesidad de copiar y pegar manualmente grandes bloques de código.



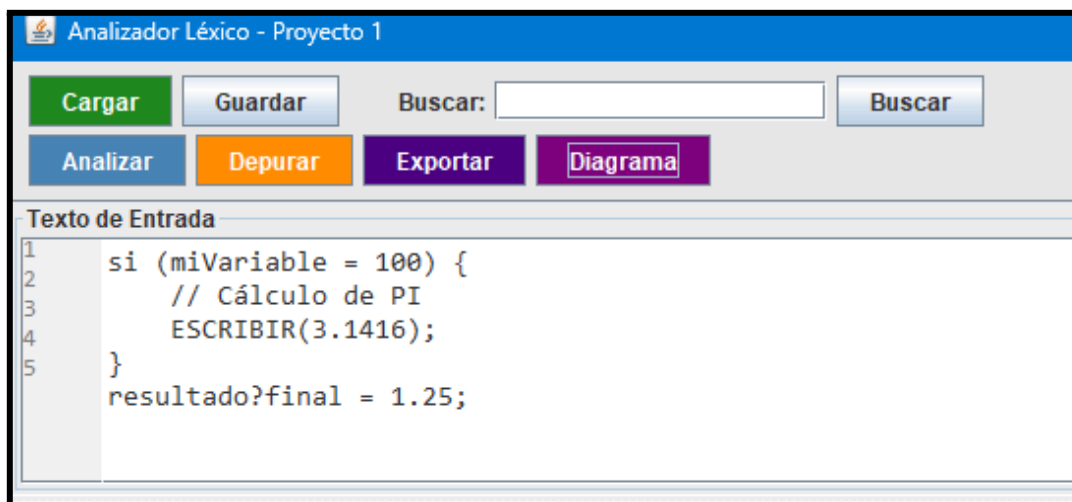
Cómo Funciona el Analizador: El Proceso Interno

Cuando usted presiona el botón "Analizar Texto", nuestra aplicación pone en marcha su Motor de Análisis Léxico. Este motor es el "cerebro" del programa, encargado de verificar que su código esté correctamente escrito según las reglas de nuestro lenguaje.

1. Preparación y Escaneo (El Motor)

Antes de comenzar a leer su código, el programa sigue estos pasos para garantizar resultados precisos:

- Limpieza de Resultados Anteriores: Si ya había analizado un texto, el motor borra automáticamente todos los reportes previos (tokens, errores, conteo) para asegurar que la nueva ejecución sea completamente limpia e independiente.
- Inicio de la Lectura: El motor empieza a leer su código desde la primera línea y la primera columna.



El motor recorre el texto carácter por carácter, desde el principio hasta el final.

- Ignora Espacios: El programa es lo suficientemente inteligente para saltar sobre espacios, tabulaciones y saltos de línea. Sin embargo, no los ignora por completo, sino que los utiliza para saber cuándo aumentar el número de columna o el número de fila, manteniendo un registro exacto de la ubicación de cada elemento de su código.

2. Reconocimiento de Patrones (El Autómata)

El motor utiliza un sistema de "Reglas de Reconocimiento" (basado en lo que llamamos un Autómata), que son como un mapa para clasificar el texto:

- Verificación Palabra por Palabra: A medida que el motor acumula caracteres, el Autómata consulta sus reglas para determinar si la secuencia actual de letras y símbolos está formando un elemento válido del lenguaje, como un Identificador, un Número o una Palabra Reservada.
- Clasificación de Elementos: Una vez que una secuencia se reconoce como un elemento completo (por ejemplo, el número 100), el Autómata lo clasifica con un Tipo de Token (en este caso, ENTERO) y registra su ubicación exacta (fila y columna).
- Identificación de Errores: Si el Autómata lee un carácter que no encaja con ninguna de sus reglas o patrones (por ejemplo, un ? en medio de un Identificador), lo marca inmediatamente como un Error y procede a reportarlo.

| Resultados del Análisis | | | |
|------------------------------------|--------------------|--------|--------|
| === REPORTE DE ANALISIS LEXICO === | | | |
| TOKENS RECONOCIDOS: | | | |
| ----- | | | |
| COMENTARIO_LINEA | '// Cálculo de PI' | Fila:2 | Col:5 |
| PALABRA_RESERVADA | 'si' | Fila:1 | Col:1 |
| IDENTIFICADOR | 'iVariable' | Fila:1 | Col:6 |
| OPERADOR | '=' | Fila:1 | Col:16 |
| ENTERO | '100' | Fila:1 | Col:18 |
| AGRUPACION | ')' | Fila:1 | Col:21 |
| AGRUPACION | '{' | Fila:1 | Col:23 |
| PALABRA_RESERVADA | 'ESCRIBIR' | Fila:2 | Col:9 |
| ENTERO | '416' | Fila:2 | Col:21 |
| AGRUPACION | ')' | Fila:2 | Col:24 |
| PUNTUACION | ' ;' | Fila:2 | Col:25 |
| AGRUPACION | ' }' | Fila:3 | Col:1 |
| IDENTIFICADOR | 'final' | Fila:4 | Col:11 |
| OPERADOR | '=' | Fila:4 | Col:17 |
| ENTERO | '1' | Fila:4 | Col:19 |
| ENTERO | '5' | Fila:4 | Col:22 |
| PUNTUACION | ' ;' | Fila:4 | Col:23 |

El motor utiliza un sistema de "Reglas de Reconocimiento" (basado en lo que llamamos un Autómata), que son como un mapa para clasificar el texto:

- Verificación Palabra por Palabra: A medida que el motor acumula caracteres, el Autómata consulta sus reglas para determinar si la secuencia actual de letras y símbolos está formando un elemento válido del lenguaje, como un Identificador, un Número o una Palabra Reservada.
- Clasificación de Elementos: Una vez que una secuencia se reconoce como un elemento completo (por ejemplo, el número 100), el Autómata lo clasifica con un Tipo de Token (en este caso, ENTERO) y registra su ubicación exacta (fila y columna).
- Identificación de Errores: Si el Autómata lee un carácter que no encaja con ninguna de sus reglas o patrones (por ejemplo, un ? en medio de un Identificador), lo marca inmediatamente como un Error y procede a reportarlo.

3. Recuperación de Errores

Una de las funciones avanzadas de nuestro analizador es la Recuperación de Errores.

- Continúa el Análisis: Si el programa encuentra un error, no se detiene. Envía el reporte del error, y luego reinicia la búsqueda del patrón para continuar escaneando el resto de su código. Esto asegura que se encuentren y se reporten todos los posibles errores en el texto, y no solo el primero que se encuentre.
- Sugerencias: En algunos casos, si el error es una palabra reservada mal escrita, la aplicación es capaz de mostrar una sugerencia de corrección.

```
Resultados del Análisis

=== REPORTE DE ERRORES CON SUGERENCIAS ===

ERROR 1:
ERROR en Fila 1, Columna 4: '(m'
  Sugerencia: Revise la sintaxis del lexema
  Sugerencia Contexto: Verifique el contexto del código

ERROR 2:
ERROR en Fila 2, Columna 17: '(3'
  Sugerencia: Revise la sintaxis del lexema
  Sugerencia Contexto: Verifique el contexto del código

ERROR 3:
ERROR en Fila 2, Columna 19: '.1'
  Sugerencia: Revise la sintaxis del lexema
  Sugerencia Contexto: Verifique el contexto del código

ERROR 4:
ERROR en Fila 4, Columna 1: 'resultado?'
  Sugerencia: Revise la sintaxis del lexema
  Sugerencia Contexto: Verifique el contexto del código

ERROR 5:
ERROR en Fila 4, Columna 20: '.2'
  Sugerencia: Revise la sintaxis del lexema
  Sugerencia Contexto: Verifique el contexto del código
```