



Universidad de Guadalajara
Centro Universitario de Ciencias Exactas e Ingenierías
Departamento de Ciencias Computacionales.
Ingeniería en Computación



Analizador Léxico

Materia: Compiladores

Sección: D08

Profesor: Hernández Andrade Jorge Fausto

Equipo:

- | | | |
|----------------------------------|--|-----------|
| ○ Cervantes Araujo Maria Dolores | | 217782452 |
| ○ Gutierrez Gachuz Fabian | | 218506653 |
| ○ Rivera Reos Fernando de Jesús | | 217882759 |

Fecha: 28/02/2025

Analizador Léxico

En esta actividad creamos la interfaz y funcionalidad principal de un compilador mediante el analizador léxico utilizando *Lark* para evaluar nuestra gramática e ir generando la tabla de tokens que posteriormente se muestra en la tabla de símbolos.

Como podemos notar, la interfaz es sencilla, simulando el entorno de un compilador como el de Visual Studio Code, contiene el editor de texto, la tabla de símbolos y la consola de errores, junto con el botón para poder compilar el algoritmo.



Escribir código: Se debe de ingresar código en el área de texto principal que va mostrando la numeración de las líneas de código que existen, en caso de que el usuario quiera compilar sin tener nada escrito sale lo siguiente en la interfaz.

Compilar: Se hace clic en el botón de compilación que lleva un diseño convencional e intuitivo para los usuarios.



Resultados: Si el usuario ingresa código al editor de texto, puede resultar en dos posibles casos:

- Si el código es válido, los identificadores se mostrarán en la "Tabla de Símbolos".
- Si hay errores, estos aparecerán en la "Consola de Errores".



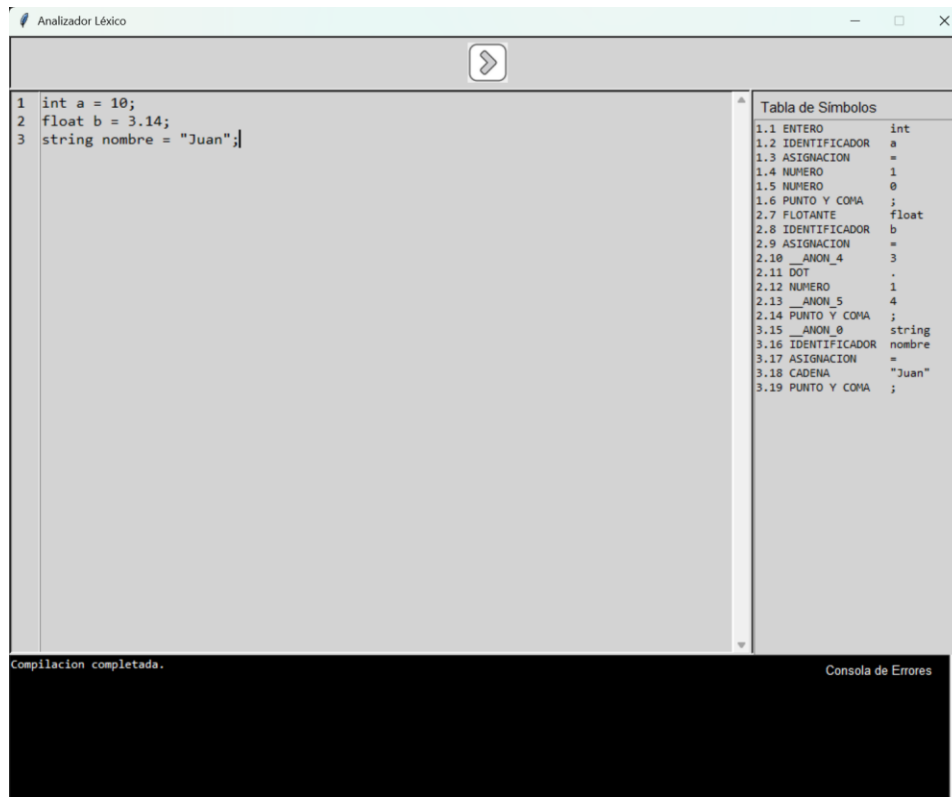
Los errores se manejan de la siguiente manera:

- **Errores de entrada vacía:** Si no hay código para compilar, se muestra un mensaje en la consola indicando que no hay símbolos que analizar.
- **Errores sintácticos:** Si el código contiene errores de sintaxis, estos se muestran en la "Consola de Errores".
- **Errores de compilación:** Si el código ingresado es incorrecto, se genera un mensaje con detalles del error.



Por lo contrario, si el algoritmo es válido en la tabla símbolos se muestra:

- Línea de código donde se encontró el identificador
- Descripción
- Nombre del identificador



En caso de que encontremos espacios en blanco, el algoritmo los ignorando y arranca desde los primeros caracteres que encuentre, ya sea código válido o inválido.



Conclusión:

Esta herramienta proporciona una interfaz gráfica intuitiva que simula un compilador, permitiendo a los usuarios ingresar código y analizarlo léxicamente. Lo que nos permite comprender mejor el funcionamiento interno de los compiladores y la importancia del procesamiento del lenguaje en la informática. A través de esta actividad, hemos aplicado principios fundamentales de análisis de código, desde la detección de tokens hasta la validación de estructuras sintácticas. Además, el uso de herramientas especializadas facilita la implementación de soluciones más eficientes y escalables. Este conocimiento no solo es relevante en el desarrollo de compiladores, sino también en otras áreas como la interpretación de datos, la seguridad informática y la automatización de procesos de software.