

# Diseño e implementación de Analizadores Léxico con lenguaje C (ó en su defecto Python y/o Java)

---

## Indicaciones

1. Proporcionar el **Prompt 1** a una IA Generativa:
2. Seguir las indicaciones generadas por la IA Gen
3. Se sugiere trabajar con el lenguaje C.
4. Descargar las herramientas señaladas en la guía proporcionada por la IA Gen.
5. Analizar y comprender: la información proporcionada, indicaciones, y los tres ejemplos proporcionados por la IA Gen.
6. Realizar los tres ejercicios sugeridos para diseñar, compilar y ejecutar los tres analizadores léxicos respectivos.
7. Proporcionar a los analizadores léxicos diversas expresiones para ser reconocidas.

## Prompt 1 (Analizadores Léxicos)

"Genera una guía técnica para que estudiantes de ingeniería en informática y computación aprendan a utilizar Lex y Flex para crear analizadores léxicos. La guía debe incluir:

Herramientas necesarias:

Para Windows: WinFlex, WSL (Windows Subsystem for Linux) o Cygwin.

Para Linux: Flex y Bison (instalables mediante `sudo apt-get install flex bison`).

Para Java: ANTLR.

Para Python: PLY (Python Lex-Yacc).

Tres ejercicios progresivos:

Ejercicio 1 (Básico): Crear un analizador léxico que reconozca palabras clave (int, return), identificadores y números en un archivo de texto. Proporcionar el código Flex, instrucciones de compilación y ejecución en Windows y Linux, y un ejemplo de salida.

Ejercicio 2 (Intermedio): Extender el analizador para reconocer comentarios (de una línea con // y de múltiples líneas con /\* ... \*/) y cadenas de texto (entre comillas dobles "..."). Incluir el código Flex actualizado y explicar cómo manejar estos casos.

Ejercicio 3 (Avanzado): Integrar el analizador léxico con un programa en C, Java o Python que cuente el número de palabras clave, identificadores, números, operadores y delimitadores en un archivo de código fuente. Proporcionar el código necesario para la integración y explicar cómo compilar y ejecutar el proyecto en Windows y Linux.

Instrucciones detalladas:

Cómo instalar y configurar las herramientas en Windows y Linux.

Cómo diseñar un archivo .l (estructura, reglas léxicas y acciones).

Cómo compilar y ejecutar el analizador léxico en cada sistema operativo.

Cómo integrar el analizador con programas en C, Java y Python.

Ejemplos de código y salidas:

Incluir ejemplos completos de archivos .l, código en C/Java/Python, y salidas esperadas para cada ejercicio.

Recomendaciones adicionales:

Cómo depurar errores comunes en el análisis léxico.

Buenas prácticas para diseñar especificaciones léxicas.

Recursos adicionales para profundizar en el tema (documentación, tutoriales, etc.)."

**Observación:** El prompt termina en la línea anterior.

# FIN

---