Implementación de un Analizador Léxico para un Nuevo Lenguaje

Entrega en equipos de máximo tres integrantes (aunque puede ser individual): 31 de marzo y 1 de abril del 2025, la recepción será durante el horario asignado a la clase.

Descripción:
El objetivo principal de este proyecto es diseñar e implementar un analizador léxico para un nuevo lenguaje de programación. El analizador léxico será capaz de reconocer y clasificar diferentes tipos de tokens, incluyendo números enteros y reales, identificadores, comentarios, palabras reservadas y operadores.
El analizador léxico también debe ser capaz de pinta los tokens de distintos colores conforme se va escribiendo código tal cual lo hacen IDE que se usan para programar.

Características Principales:

1. Reconocimiento de Tokens:

- Números enteros y reales (con y sin signo) (Color 1)
- Identificadores compuestos por letras y dígitos sin comenzar por dígito(Color 2)
- Comentarios de una línea y de múltiples líneas al estilo del lenguaje C (Color 3)
- Palabras reservadas: if, else, end, do, while, switch, case, int, float, main, cin, cout (Color 4)
- Operadores aritméticos: +, -, *, /, modulo(%), potencia (^), ++ (incremento), -- (decremento) (Color 5)
- Operadores relacionales: <, <=, >, >=, !=, == (Color 6, todos los operadores llevan el mismo color)
- Operadores lógicos: and(&&), or(||) y not(!=). (Color 6, todos los operadores llevan el mismo color)
- Símbolos: (,),{, }, coma(,), punto y coma (;)
- Asignación: =

2. Implementación del Analizador Léxico:

- Utilizar un lenguaje de programación de tu elección (por ejemplo, Python, go, etc.)
- Diseñar un algoritmo eficiente para reconocer y clasificar los tokens mencionados anteriormente
- Implementar la lógica necesaria para manejar comentarios de una línea y de múltiples líneas
- Definir cómo manejar los errores léxicos, como caracteres inválidos o tokens no reconocidos, debe indicar errores con número de línea y de columna.

3. Pruebas y Validación:

- Desarrollar un conjunto de pruebas exhaustivas para verificar el correcto funcionamiento del analizador léxico

- Validar el analizador léxico utilizando ejemplos de código representativos que contengan todos los tipos de tokens especificados

4. Documentación:

- Documentar el diseño y la implementación del analizador léxico, explicando cada componente y algoritmo utilizado
- Proporcionar instrucciones claras sobre cómo compilar y ejecutar el analizador léxico

Entregables:

- Autómata de estado finito
- Código fuente del analizador léxico
- Conjunto de pruebas exhaustivas
- Documentación detallada del proyecto
- Archivo de tokens desplegado en la ventana del IDE diseñado para tal fin
- Archivo de errores desplegado en la ventana del IDE para tal fin.
- Entrega en aula virtual en un archivo comprimido, solo uno por equipo, incluyendo un block de notas con los nombres de los integrantes del equipo.

Consideraciones Adicionales:

- Se recomienda utilizar herramientas y técnicas estándar de análisis léxico, como expresiones regulares y autómatas finitos deterministas (DFA).
- Es importante mantener un diseño modular y bien estructurado para facilitar el mantenimiento y la escalabilidad del analizador léxico.