

Primer programa
Analizador Léxico**Objetivo**

Elaborar un analizador léxico en *lex/flex* que reconozca los componentes léxicos pertenecientes a las clases abajo descritas y que fueron definidas en clase.

Descripción

- Las clases de los componentes léxicos válidos para el analizador léxico son:

Clase	Descripción
0	Palabras reservadas (ver tabla)
1	Símbolos especiales (ver tabla)
2	Identificadores. Deben empezar por @, le deben seguir uno o más de los siguientes caracteres de A-Z y a-z, vocales mayúsculas y minúsculas pueden ir acentuadas, dígitos del 0 al 9, acepta ñ y Ñ; y debe terminar con un guion bajo _.
3	Operadores aritméticos (ver tabla)
4	Operadores relacionales (ver tabla)
5	Operadores de asignación (ver tabla)
6	Constantes cadena. Inician y terminan con -- . Pueden contener cualquier carácter.
7	Constantes enteras. No pueden iniciar con = si son 2 o más dígitos. Pueden tener signo + o - . Puede tener el sufijo p (pequeño) o g(grande). Ejemplos: 0 -24 +34p 18g
8	Constantes reales: La parte fraccionaria se escribe después de ' (apóstrofo). Ejemplos: 12'2 242'87 '604 Para escribir un valor real sin fracciones se pueden usar los sufijos r o R. Ejemplos: 23r 649R

El número de clase es inamovible.

- El analizador léxico tendrá como entrada un archivo con el programa fuente, **el cual se indicará desde la línea de comandos** al momento de mandar a ejecutar el analizador léxico.
- Como delimitador de un componente léxico será uno o varios espacios, tabuladores o saltos de línea, **así como el inicio de otro componente léxico**. Considerar el orden en las acciones en el programa *lex/flex*
- Los tokens se representarán en una estructura con dos campos:
campo1: la clase
campo2: el valor (de acuerdo con las siguientes tablas e indicaciones)

Valor	Palabra reservada	Valor	Palabra reservada
0	Bool	8	Haz
1	Cade	9	Mientras
2	Continuar	10	Nulo
3	Devo	11	Para
4	Ent	12	Parar
5	Fals	13	Si
6	Flota	14	Sino
7	Global	15	Ver

Valor	Op. aritmético
0	sum
1	rest
2	mult
3	div
4	mod
5	inc
6	dec
7	exp
8	dive

Valor	Op. asignación
0	->
1	+->
2	-->
3	*->
4	/->
5	%->
6	>>
7	<<
8	^->
9	&->

Valor	Símbolo Especial	Representa
0	<	(
1	>)
2	<<	[
3	>>]
4	#	{
5	#!	}
6	*	;
7		,
8	°	.

Valor	Op. relacional	Representa
0	h	Mayor que
1	m	Menor que
2	e	Igual que
3	c	Diferente a
4	he	Mayor o igual
5	me	Menor o igual

El valor para el token de cada identificador es la posición dentro de la tabla de símbolos. Para las palabras reservadas, los símbolos especiales, los operadores aritméticos, los operadores relacionales y los operadores de asignación, será la posición en su correspondiente tabla (catálogo).

Las constantes cadenas se incluirán en una tabla de literales, al igual que las constantes flotantes, por lo que el valor de su token será la posición dentro de su correspondiente tabla.

Cada vez que el Analizador Léxico encuentre una cadena o una constante flotantes NO revisará si ya se encuentra en su tabla de literales, simplemente la insertará.

Las constantes enteras como valor será el valor numérico de la constante entera.

Cuando detecte un error léxico, deberá seguir el reconocimiento a partir del siguiente símbolo.

- El analizador deberá crear la Tabla de Símbolos para almacenar a los identificadores. Esta tabla manejará los campos: posición, nombre del identificador y tipo (este último será de tipo entero y podrán ponerle como valor inicial -1). Se indicará en el documento a entregar, el tipo de estructura de datos empleada, así como el método de búsqueda a utilizar.
- Las tablas de literales para las cadenas y para las constantes flotantes, deberán tener como estructura, dos campos: la posición y el dato (cadena).
- El Analizador Léxico deberá reconocer los comentarios y descartarlos. Los comentarios se identifican porque estarán entre [] (corchetes).

- Al término del análisis léxico deberá mostrar la tabla de símbolos, las tablas de literales, así como la secuencia de tokens. También podrán almacenarse en archivos para su mejor revisión.
- Los errores que vaya encontrando el analizador léxico, los podrá ir mostrando en pantalla o escribirlos en un archivo. Deberá recuperarse de los errores encontrados para continuar con el reconocimiento de todos los componentes léxicos del archivo de entrada.
- El programa deberá estar comentado, con una descripción breve de lo que hace (puede ser el objetivo indicado en este documento), el nombre de quién(es) elaboró(aron) el programa y fecha de elaboración, así como lo que hace cada función. Se deberá cuidar mucho la sangría que denota la dependencia de instrucciones.

Entregar:

Un documento con la siguiente estructura:

- Descripción del problema (no del programa), incluyendo características que deben cumplir los componentes léxicos que reconocerá el analizador léxico y **la expresión regular de cada clase**.
- Propuesta de solución y fases del desarrollo del sistema: Análisis (planificación, indicando participantes por cada actividad), Diseño e Implementación. En el diseño indicar exactamente cómo se definirán la tabla de símbolos, las tablas de literales y los tokens, la técnica de búsqueda e inserción de los identificadores.
- Indicaciones de cómo correr el programa.
- Conclusiones por cada participante.

Nota: se podrá elaborar individualmente o en equipo de 2

Enviar el documento y sólo el programa fuente definitivo a la plataforma educativa (apartado tareas y empaquetados en .rar o .zip) y en su caso, **sólo un miembro del equipo**.

Fecha de entrega: 18 de marzo de 2025.