

# Taller 1 Analizador Léxico Lenguaje SL

**Grupos:** 2 personas por grupo

El objetivo para esta práctica es tomar un código fuente escrito en el lenguaje SL y realizar un análisis léxico sobre dicho código.

## Entrada - Salida

La entrada para juzgar el analizador léxico está dada por un archivo que contiene código fuente escrito en lenguaje SL, que su programa debe analizar a nivel léxico para entregar la salida adecuada. Por ejemplo:

Entrada	Salida
<pre>// Comentario var     n, suma, k : numerico  inicio     imprimir ("Suma pares hasta n.", " Ingrese n: ")     leer (n)     suma = 0.0     desde k=2 hasta n paso 2     {         suma = suma + k     }     imprimir ("La suma es ", suma) fin</pre>	<pre>&lt;var,2,1&gt; &lt;id,n,3,5&gt; &lt;tk_coma,3,6&gt; &lt;id,suma,3,8&gt; &lt;tk_coma,3,12&gt; &lt;id,k,3,14&gt; &lt;tk_dospuntos,3,16&gt; &lt;numerico,3,18&gt; &lt;inicio,5,1&gt; &lt;imprimir,6,5&gt; &lt;tk_par_izq,6,14&gt; &lt;tk_cadena,"Suma pares hasta n.",6,15&gt; &lt;tk_coma,6,36&gt; &lt;tk_cadena," Ingrese n: ",6,37&gt; &lt;tk_par_der,6,51&gt; &lt;leer,7,5&gt; &lt;tk_par_izq,7,10&gt; &lt;id,n,7,11&gt; &lt;tk_par_der,7,12&gt; &lt;id,suma,8,5&gt; &lt;tk_asig,8,10&gt; &lt;tk_num,0.0,8,12&gt; &lt;desde,9,5&gt; &lt;id,k,9,11&gt; &lt;tk_asig,9,12&gt; &lt;tk_num,2,9,13&gt; &lt;hasta,9,15&gt; &lt;id,n,9,21&gt; &lt;paso,9,23&gt; &lt;tk_num,2,9,28&gt; &lt;tk_llave_izq,10,5&gt; &lt;id,suma,11,9&gt; &lt;tk_asig,11,14&gt; &lt;id,suma,11,16&gt; &lt;tk_suma,11,21&gt; &lt;id,k,11,23&gt; &lt;tk_llave_der,12,5&gt; &lt;imprimir,13,5&gt; &lt;tk_par_izq,13,14&gt; &lt;tk_cadena,"La suma es ",13,15&gt; &lt;tk_coma,13,28&gt; &lt;id,suma,13,30&gt; &lt;tk_par_der,13,34&gt; &lt;fin,14,1&gt;</pre>

Del ejemplo anterior, se pueden generalizar las reglas para realizar el análisis léxico:

- La salida consiste en una lista de tokens separados por saltos de línea, los cuales deben seguir el siguiente formato **<tipo\_de\_token,lexema,fila,columna>** . En el caso de las palabras reservadas, el token y el lexema son iguales. Entonces se debe imprimir el siguiente formato **<tipo\_de\_token,fila,columna>**. Queda como parte del ejercicio encontrar el conjunto de palabras reservadas para SL, según la especificación dada en el manual de referencia.
- Se deben especificar los lexemas de la siguiente manera, dependiendo de su token (recordar que los únicos valores que pueden ir fuera de comillas son los numéricos).

**<tk\_cadena,lexema,fila,columna>**

**<tk\_num,lexema,fila,columna>**

**<tk\_logico,lexema,fila,columna>**

**El lexema del número puede ser entero o real (con o sin signo)**

- Los identificadores encontrados seguirán el siguiente formato:

**<id,lexema,fila,columna>**,

- Los comentarios deberán ser ignorados.
- Tenga en cuenta que SL es case sensitive, por lo que una cadena como NUMerico no será aceptada como palabra reservada. En cambio, la palabra numerico si es considerada como palabra reservada.
- Los operadores y símbolos especiales deben imprimirse en el siguiente formato:

**<nombre\_token,fila,columna>**

Donde nombre\_token será determinado de la siguiente manera: tk\_nombre\_token. El nombre\_token puede ser asignado libremente según su criterio. En el ejemplo anterior se utilizaron, por ejemplo: tk\_coma, tk\_dospuntos, tk\_par\_izq, tk\_par\_der, tk\_llave\_izq, tk\_llave\_der, tk\_asig, tk\_suma, ...

## Errores

Si durante el proceso se detecta un error léxico se deberá abortar el análisis y se debe reportar el error de la siguiente manera:

**>>> Error lexico(linea:X,posicion:Y)**

Donde X y Y son números enteros que representan el número de línea y la posición donde fue detectado el inicio del error léxico.

## Nota

- Si en la entrada de texto se detecta una cadena como esta 90.00.50, el analizador deberá detectar un tk\_num 90.00, luego detectar el tk\_punto, y por último, el tk\_num 50. Esto ocurre porque el "." podría aparecer como un token único en este lenguaje; de lo contrario, sería un error léxico. En este caso no es un error léxico.
- Si se detecta por ejemplo la cadena 1&23948998, el analizador deberá detectar el número 1 y luego marcar un error léxico en el &.
- Recuerde tener en cuenta casos extremos como por ejemplo: 2.5598055not3<>88&56.a , en este caso se detecta la siguiente secuencia:

```
<tk_num,2.5598055,1,1>  
<id,not3,1,10>  
<tk_distinto,1,14>  
<tk_num,88,1,16>  
>>> Error lexico(linea:1,posicion:18)
```

- A la hora de imprimir una palabra reservada como "desde", debe imprimir la misma palabra reservada en el espacio de token <desde,fil,columna>.
- A la hora de realizar el análisis léxico se deberá seguir el principio de subcadena más larga, a continuación un ejemplo:

<>\$	<tk_distinto,1,1> >>> Error lexico(linea:1,posicion:3)
------	---

Como se puede ver el análisis toma el token '<>' pues es el token más largo que puede tomar, y con el substring restante '\$' informa un error léxico pues este carácter no está definido dentro de los tokens válidos para el lenguaje.