



Sección: D02 Profesor: MICHEL EMANUEL LOPEZ FRANCO
Tema: Otras herramientas para el manejo de errores Ejemplos
Ciclo: 2024A

Como un ejemplo minimo de lo que es un analizador léxico vemos el siguiente código como un ejemplar para este modelo debido a que será la conjunción de un compilador completo:

```
1  #include <iostream>
2  #include <fstream>
3  #include <regex>
4
5  using namespace std;
6
7  string datoToken;
8  bool buscarEnArchivo(const string &token)
9  {
10     ifstream tokensFile;
11     string tok, def, line;
12     bool exist=false;
13     tokensFile.open("tokens.txt");
14     if(!tokensFile.good())
15     {
16         cout<<"Archivo no encontrado.";
17     }
18     else
19     {
20         while(!tokensFile.eof())
21         {
22             getline(tokensFile, tok, '@');
23             getline(tokensFile, def, '\n');
24
25             if(tok==token)
26             {
27                 datoToken=def;
28                 exist=true;
29             }
30         }
31     }
32     tokensFile.close();
33     return exist;
34 }
```

Ya que en esta imagen se puede apreciar a simple vista el como esta usando las definiciones de lo que necesitamos que este mismo detecte se procede a analizar la parte de la funcionalidad como en la siguiente imagen.

```
parentesis      )  
llave           {  
identificador    4  
identificador    0  
operador         =  
entero           1  
punto_y_coma     ;  
identificador    4  
identificador    0  
operador         =  
real             2  
punto_y_coma     ;  
identificador    19  
parentesis      (  
identificador    0  
operador         >
```

En el cual esta viene siendo la salida aunque esta configurado de manera en que la salida es automática y por lo tanto se nota como solo interpreta lo que se ha ingresado.

Como excepciones y algunos datos lógicos se tienen las siguientes tablas para la referencia de funcionalidad.

Identificadores= letra (letra | dígito)*

Entero= dígito⁺

Real= entero.entero

Operador de adición: + | -

Operador de multiplicación: * | /

Operador de asignación: =

Operador relacional: < | > | <= | >= | != | ==

Operador And: &&

Operador Or: ||

Operador Not: !

Parentesis: (,)

Llave: { , }

Punto y coma: ;

Además de las siguientes palabras reservadas: **if, while, return, else, int, float**

Símbolo	Tipo	
identificador	0	
entero	1	
real	2	
cadena	3	
tipo	4	int,float,void
opSuma	5	+, -
opMul	6	*, /
opRelac	7	<, <=, >, >=
opOr	8	
opAnd	9	&&
opNot	10	!
opIgualdad	11	==, !=
;	12	
,	13	
(14	
)	15	
{	16	
}	17	
=	18	
if	19	
while	20	
return	21	