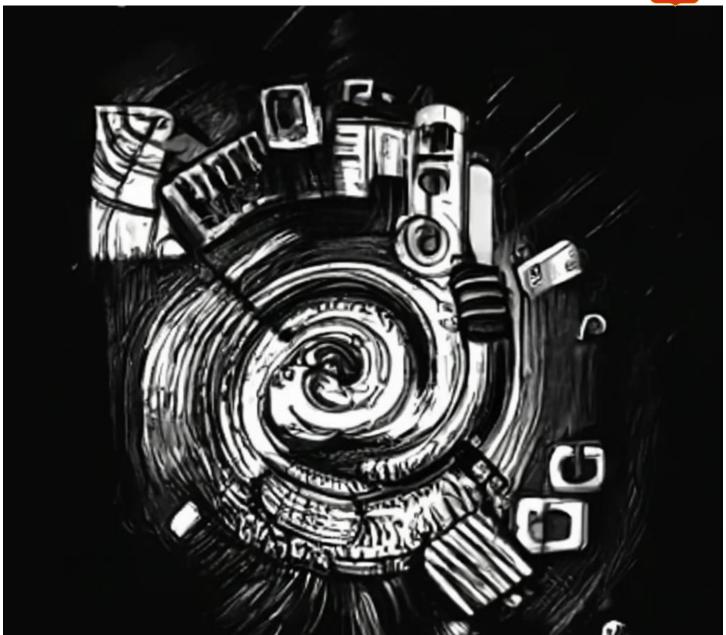




Universidad de Guadalajara Centro Universitario de Ciencias Exactas e Ingenierías





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:

```
#include <iostream>
#include <fstream>
#include <regex>
using namespace std;
string datoToken;
bool buscarEnArchivo(const string &token)
    ifstream tokensFile;
    string tok, def, line;
    bool exist=false;
    tokensFile.open("tokens.txt");
    if(!tokensFile.good())
        cout<<"Archivo no encontrado.";</pre>
    else
        while(!tokensFile.eof())
            getline(tokensFile, tok, '@');
            getline(tokensFile, def, '\n');
            if(tok==token)
                datoToken=def;
                exist=true;
    tokensFile.close();
    return exist;
```

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.

IIVCO		
parencesis)
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		Ò
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|digito)*

Entero= digito<sup>†</sup>

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

T	
Tipo	
0	
1	
2	
3	
4	int,float,void
5	+,-
6	*,/
7	<,<=,>,>=
8	11
9	&&
10	!
11	==, !=
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20