



Nombre: Dana Carolina Ramírez Velázquez

Código: 220286547

Materia: Traductores de Lenguaje 2

Actividad: Parte 1 Proyecto

Fecha: 23/11/22

Introducción:


Las indicaciones para elaborar la primera parte del proyecto eran implementar un analizador sintáctico descendente de la gramática del lenguaje de programación corregido en la práctica 3.

Desarrollo:

La gramática era la siguiente:

```
<programa>      → begin <declaraciones> <órdenes> end  
<declaraciones> → <declaración> ; <sig_declaraciones>  
<sig_declaraciones> → <declaración> ; <sig_declaraciones> |  $\epsilon$   
<declaración>   → <tipo> <lista_variables>  
<tipo>          → entero | real
```

```
<lista_variables> → identificador <sig_lista_variables>  
<sig_lista_variables> → , <lista_variables> |  $\epsilon$   
<órdenes>        → <orden> ; <sig_órdenes>  
<sig_órdenes>     → <orden> ; <sig_órdenes> |  $\epsilon$   
<orden>           → <condición> | <bucle_while> | <asignar>  
<condición>       → if ( <comparación> ) <órdenes> <sig_condición>  
<sig_condición>   → end  
                  | else <órdenes> end
```


C AnalizadorLexico.h >  AnalizadorLexico

```
3  #include <iostream>
4  #include <string>
5  #include <vector>
6  #pragma once
7  #include <fstream>
8
9  #include "Hoja.h"
10 using namespace std;
11 enum class TokenType {
12     ESPACIO,
13     NUMERO,
14     IDENTIFICADOR,
15     BEGIN,
16     END,
17     TIPO,
18     COMA,
19     PUNTO,
20     ASIGNACION,
21     PUNTO_Y_COMA,
22     DOS_PUNTOS,
23     PARENTESIS,
24     PARENTESIS_END,
25     BUCLE,
26     BUCLE_END,
27     CONDICION,
28     CONDICION_ELSE,
29     OPERADOR_BOOL,
30     OPERADOR_ARITMETICO,
31 };
```

```

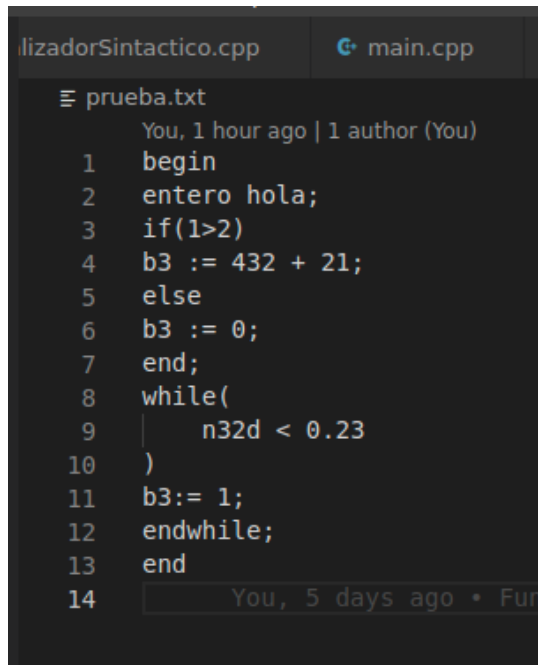
33  class AnalizadorLexico {
34      private:
35          fstream Archivo;
36          string linea;
37          string _token;
38
39          void leer();
40          bool isNumber();
41          void separate();
42          void evaluate();
43
44      public:
45          struct Token {
46              TokenType tipo;
47              string value;
48              int id;
49          };
50
51          int total = 0;
52          AnalizadorLexico();
53          vector<Hoja> Tabla;
54          vector<Token> List;
55          int LookTable(string valor);
56          void start(string file);
57      };
58
59      #endif

```

En la definicion de las funciones empieza a mandar mensajes de error en caso de que la entrada no sea la información que se esperaba, asímismo podemos ver que por temas de practicidad, lo que recibe de entrada es un archivo de texto.

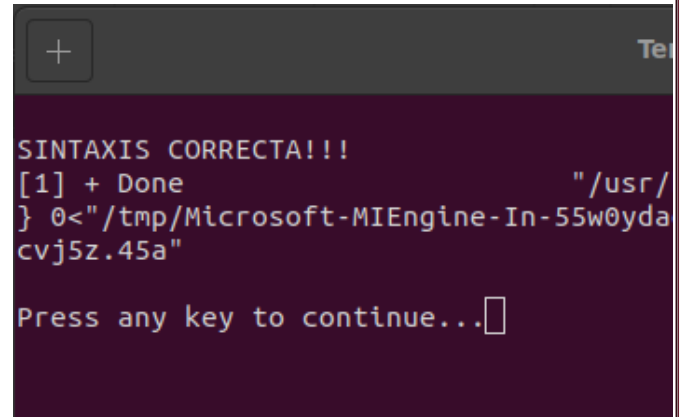
Resultados:

Intendando con la siguiente entrada:

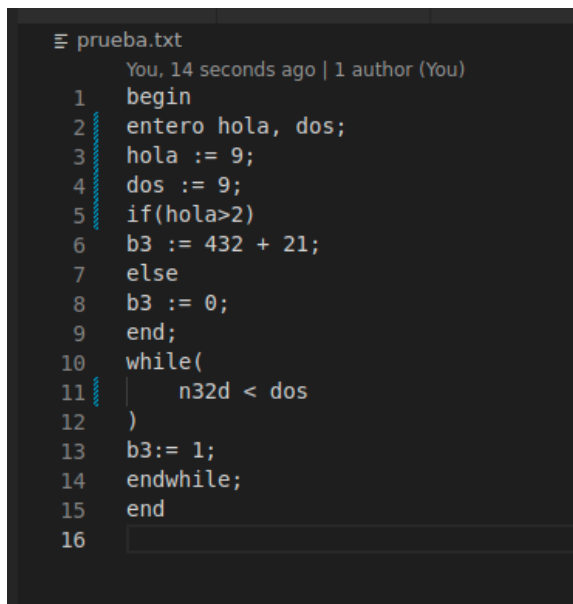


```
lizadorSintactico.cpp  main.cpp
prueba.txt
You, 1 hour ago | 1 author (You)
1 begin
2 entero hola;
3 if(1>2)
4 b3 := 432 + 21;
5 else
6 b3 := 0;
7 end;
8 while(
9     n32d < 0.23
10 )
11 b3:= 1;
12 endwhile;
13 end
14 You, 5 days ago • Fur
```

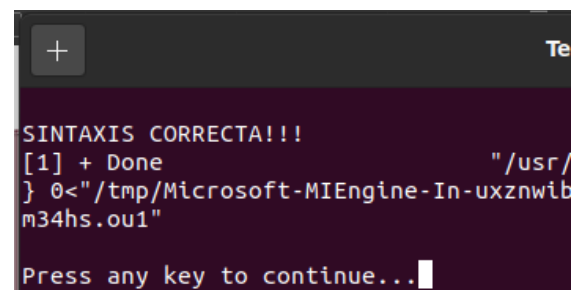
El resultado es el siguiente:



```
+ Ter
SINTAXIS CORRECTA!!!
[1] + Done "/usr/
} 0<"/tmp/Microsoft-MIEngine-In-55w0yda
cvj5z.45a"
Press any key to continue...
```



```
prueba.txt
You, 14 seconds ago | 1 author (You)
1 begin
2 entero hola, dos;
3 hola := 9;
4 dos := 9;
5 if(hola>2)
6 b3 := 432 + 21;
7 else
8 b3 := 0;
9 end;
10 while(
11     n32d < dos
12 )
13 b3:= 1;
14 endwhile;
15 end
16
```



```
+ Te
SINTAXIS CORRECTA!!!
[1] + Done "/usr/
} 0<"/tmp/Microsoft-MIEngine-In-uxznwib
m34hs.ou1"
Press any key to continue...
```

Entrada Erronea:

Mensaje resultado:

ctico.cppmain.cppHoja.h 3stl_vector.hprueba.txt

prueba.txt

You, 5 seconds ago | 1 author (You)

1begin

2entero var1,var2;

3real var3;

4var1 := 3

5var2 := 4 +(3-2);

6if(var1>var2)

7var3 := var1 + (21 - 30) + 9;

8var2 := 5;

9else

10var2 := 1;

11end;

12var1 := 23;

13while(var1<var2)

14var3 := var1;

15var2 := 9;

16var1 := 4+2+23;

17var2 := var3;

18endwhile;

19var1 := 777 - 999;

20end

+

Terminal

sintaxis incorrecta unexpected token: var2
[1] + Done "/usr/bin/gdb
> 0<"/tmp/Microsoft-MIEngine-In-qcrynoid.vgq"
jxxf1.xz0"
Press any key to continue...

prueba.txt

You, 8 seconds ago | 1 author (You)

1begin

2entero var1,var2;

3real var3;

4var1 := 3;

5var2 := 4 +(3-2);

6if(var1>var2)

7var3 := var1 + (21 - 30) + 9;

8var2 := 5;

9else

10var2 := 1;

11end;

12var1 := 23;

13while

14var3 := var1;

15var2 := 9;

16var1 := 4+2+23;

17var2 := var3;

18endwhile;

19var1 := 777 - 999;

20end

+

Terminal

sintaxis incorrecta unexpected token: var3
[1] + Done "/usr/bin/gdb
> 0<"/tmp/Microsoft-MIEngine-In-icmykkri.a0o"
gxz4e.o10"
Press any key to continue...

Conclusiones:

Esta fue la parte más sencilla del proyecto, pues solo constaba de traducir la gramática a funciones recursivas en el lenguaje que estaba utilizando, que era C++ y adecuar el analizador Léxico que previamente hice en las practicas anteriores para que funcionara con los tokens validos en esta ocasión.

Código indentado: