

PROYECTO. ANALIZADOR LÉXICO

Desarrollo: Individual, en el Lenguaje de Programación que usted elija.

Entrega: Enviar la carpeta del proyecto (comprimida como .zip) a ginobarroso@uagrm.edu.bo hasta el Vie 05/diciembre/2025, 23:59:59 hrs.

Defensa: Vie 12/diciembre/2025, 11:30 am, en el Lab 236-45

Importante: No se olvide de llevar el dibujo de su Diagrama de Transiciones (dt).

Desarrollar un Analizador Léxico (Analex) para un lenguaje que toma características de JAVA y Pascal, tomando en cuenta que el mismo **NO** es **case-sensitive**. Los tokens a reconocer son:

Ctte del Nombre	TOKEN	LEXEMA (S)
0	< FIN,_>	" "
1	< ERROR,_>	(El lexema que dio el error)
2	< VAR,_>	"var"
3	< ARRAY,_>	"array"
4	< VOID,_>	"void"
5	< IF,_>	"if"
6	< ELSE,_>	"else"
7	< TO,_>	"to"
8	< DO,_>	"do"
9	< WHILE,_>	"while"
10	< READLN,_>	"readln"
11	< PRINTLN,_>	"println"
12	< UNTIL,_>	"until"
13	< FALSE,_>	"false"
14	< TRUE,_>	"true"
15	< RETURN,_>	"return"
16	< MAIN,_>	"main"
17	< COMA,_>	", "
18	< PTOCOMA,_>	"; "
19	< CA,_>	" ["
20	< CC,_>	"] "
21	< PA,_>	" ("
22	< PC,_>	") "
23	< LA,_>	" { "
24	< LC,_>	" } "
25	< POR,_>	" * "
26	< MOD,_>	" % " y "mod"
27	< MAS,_>	" + "
28	< MENOS,_>	" - "
29	< DIV,_>	" / " y "div"
30	< NOT,_>	" ! " y "not"
31	< AND,_>	"and"

Ctte del Nombre	TOKEN	LEXEMA (S)
32	<OR,_>	"or"
33	<DOTS,_>	".."
34	<DOSPUNTOS,_>	
35	<ASSIGN,_>	"=" y ":"
36	<NUM, valor>	(Números enteros sin signo)
37	<ID, -1>	(Empiezan con Letra y luego le sigue una combinación de Letras y/o Dígitos)
38	<STRINGctte, 0>	(Chars encerrados entre comillas "...", sin EOLN ni EOF)
39	<OPREL,IGUAL> <OPREL,MEN> <OPREL,MAY> <OPREL,MEI> <OPREL,MAI> <OPREL,DIS>	"=" "<" ">" "<=" ">="
40	<TIPO, CHAR> <TIPO, BOOLEAN> <TIPO, INT>	"char" "boolean" "integer"

Puesto que su Analex no trabajará con la Tabla de Símbolos, para el token **ID** use como atributo **-1** , y para el **STRINGctte**, el 0: **<ID, -1>, <STRINGctte, 0>**.

Comentarios. Los comentarios que su Analex, debe tratar son:

- Comentarios de línea: //...
- Comentarios multilínea: /* ... */

OBLIGADO. Usted debe implementar una **TPC** –Tabla de Palabras Claves (Keywords)–, para reconocer las palabras reservadas del lenguaje. Recuerde que la TPC es una tabla **private** de la class Analex.

Las Keywords de este lenguaje son 18:

```
var, array, void, if, else, to, do, while, readln, println, until, false,
true, return, main, char, boolean, integer
```

```

//Ejemplo de un programa, escrito en el lenguaje.
//El lenguaje NO es case-sensitive.

var
  x, y, n : Integer;
  b : BOOLEAN;
  a, c : Char; /* Solo se manejan vars globales y los tipos Integer, Boolean y
                  Char */
  v : Array[1..10] Integer;
```



```

void lectura() {
  b = true;
  do
    println("Introduzca N: ");
    Readln(N);
    b := (N >= 0);
    if (!b)
      println("El valor ", N, " es incorrecto");
    else
      return;
  until (false);
}
```



```

var
  f, i : Integer;
```



```

void factorial() {
  f := 1;
  i := N;
  while i>0 do{
    f = f * i;
    i = i - 1;
  }
}
```



```

void factorial2() {
  f = 1;
  while i<=n do{
    f := f * i;
    i := i + 1;
  }
}
```



```

main(){ //main es una palabra reservada (keyword).
  lectura();
  factorial();
  println("El factorial de ", N, " es ", f);
}
```