# PROCESADORES DE LENGUAJES





# Grupo 64

Christian Paniagua Paniagua x150380

Javier Pérez Martín x150147

Pablo Heras Aranzana x150046

# <u>Índice</u>

1.	. 1	Introducción	5
2.	. [	Diseño Analizador Léxico	5
	2.1	1. Tokens	5
	2.2	2. Gramática	6
	2.3	3. Autómata	7
	2.4	4. Acciones semánticas	8
3. Diseño Analizador Sintáctico			9
	3.1	1. Gramática	9
	3.2	2. Demostración LL(1)	10
	3.3	3. Analizador descendente recursivo	11
4.	. [	Diseño Analizador Semántico	15
5.	. 1	Tabla de símbolos	20
6.	. (	Gestor de errores	20
Α	nex	хо	22

#### 1. Introducción

En esta memoria se detalla el diseño del compilador de Javasctipt-PL de la asignatura Procesadores de lenguajes 2018-2019. La implementación se ha realizado en el lenguaje Python en su versión 3. Para ejecutar el programa debe tenerse instalado Python 3 y ejecutar desde la carpeta src el comando 'python sintacticosemantico.py'. Posteriormente el programa pedirá al usuario introducir la dirección relaitva a src del fichero con el código fuente a compilar. Los ficheros de salida se almacenarán en la carpeta Salida después de la ejecución.

#### 2. Diseño Analizador Léxico

El analizador léxico se ha implementado usando la herramienta PLY para Python. El diseño es el siguiente:

#### 2.1. Tokens

<cteent, número=""></cteent,>	< <b>OPMAY,</b> - > (>)
<cad, "lexema"=""></cad,>	<opmen, -=""> (&lt;)</opmen,>
<ctebool, 1=""> (true)</ctebool,>	< <b>OPIG, -&gt;</b> (==)
<ctebool, 0=""> (false)</ctebool,>	< <b>OPDISTINTO,-&gt;</b> (!=)
<mas, -=""> (+)</mas,>	<dospuntos, -=""> (:)</dospuntos,>
<menos, -=""> (-)</menos,>	< <b>PR, 1 &gt;</b> (int)
<mul, -=""> (*)</mul,>	< <b>PR, 2 &gt;</b> (bool)
<div, -=""> (/)</div,>	< <b>PR, 3 &gt;</b> (string)
<mod, -=""> (%)</mod,>	< <b>PR, 4 &gt;</b> (if)
< <b>AND,</b> -> (&&)	< <b>PR, 5 &gt;</b> (default)
<llava, -=""> ( { )</llava,>	< <b>PR, 6 &gt;</b> (break)
<llavc, -=""> ( } )</llavc,>	< <b>PR, 7 &gt;</b> (return)
<para, -=""> ( ( )</para,>	<pr, 8=""> (function)</pr,>
< <b>PARC, -&gt;</b> ())	< <b>PR, 9 &gt;</b> (var)
< <b>FIN,</b> - > (;)	< <b>PR, 10 &gt;</b> (switch)
<sig, -="">(,)</sig,>	< <b>PR, 11 &gt;</b> (case)
<or, -=""> (  )</or,>	< <b>PR, 12 &gt;</b> (print)
<not, -=""> (!)</not,>	< <b>PR, 13 &gt;</b> (prompt)
<asigr, -=""> (-=)</asigr,>	<ld><ld, pos_ts=""> (siendo "pos_ts" la posición</ld,></ld>
<asig, -=""> (=)</asig,>	que ocupa el id dentro de la tabla de símbolos)

#### 2.2. Gramática

```
S --> del S | d E | 'C | | | | | | N | = D | : | + | - D1 | & J | ' | ' K | * | [ | ] | ; | , | ! D2 | < D3 | D4 > C --> \ C' | oc1 C | '

C'--> 't' C | 'n' C | 'C | C | C

D --> = | \lambda

D1 --> = | \lambda

D2 --> = | \lambda

D4 --> = | \lambda

E --> d E | \lambda

I --> | | | d | | _ | | \lambda

J --> &

K --> '| '

N --> * N' | \lambda

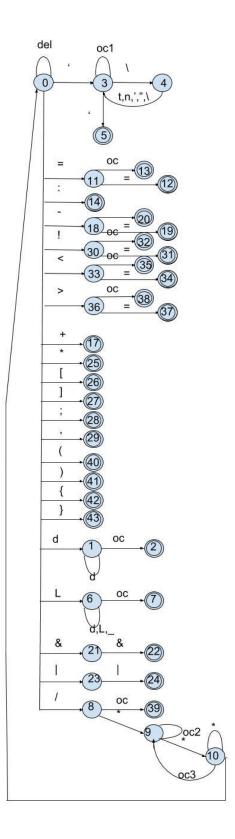
N' --> * N'' | oc2 N'

N''--> * N'' | / S | oc3 N'
```

#### Definiciones:

```
del={esp, tab, eol}
d = {0..9}
l = {a..z, A..Z}
oc1 = {cualquier carácter imprimible} - {\}
oc2 = {cualquier carácter imprimible} - {*}
oc3 = {cualquier carácter imprimible} - {/,*}
```

# 2.3. Autómata



#### 2.4. Acciones semánticas

```
L=Leer_carácter()
                                                          10:10-->L
0:1 -->L, valor = valor ascii(d)
                                                          10:0-->L
1:1 -->L, valor = valor*10 + valor ascii(d)
                                                          0:11-->L
1:2 -->if( valor > 32767) then ERROR
                                                          11:12-->L, G.token(<OP_IG, pos>)
       else G.Token(<ENT, valor>)
                                                          11:13-->G.token(<Asig, ~>)
0:3 -->L,lexema = carácter ascii(')
                                                         0:14-->L, G.token(<Dos_puntos, ~>)
3:3 -->L,lexema = lexema ⊕ carácter ascii(oc1)
                                                         0:17-->L, G.token(<Mas, ~>)
3:4 -->L,lexema = lexema ⊕ carácter ascii(\)
                                                         0:18-->L
4:3 --> L,lexema = lexema ⊕ {
                                                          18:19-->L, G.token(<Asig_R, ~>)
                                                          18:20--> G.token(<Menos, ~ >)
      carácter ascii(t)
      carácter ascii(n)
                                                          0:21-->L
      carácter ascii( ')
                                                          21:22-->L, G.token(<AND, ~>)
      carácter ascii(")
                                                         0:23-->L
                                                          23:24-->L, G.token(<OR, ~>)
      carácter ascii(\)
                                                         0:25-->L, G.token(<MUL, ~>)
    }
3:5-->L, lexema = lexema ⊕ carácter ascii( ')
                                                         0:26-->L, G.token(<CORA, ~>)
0:6-->L, lexema = carácter asciii( I )
                                                         0:27-->L, G.token(<CORC, ~ >)
6:6-->L, lexema = lexema ⊕ {
                                                         0:28-->L, G.token(<FIN, ~>)
      carácter ascii(I)
                                                         0:29-->L,
      carácter ascii(d)
                                                         0:30-->L
      carácter ascii( -)
                                                          30:31-->L, G.token(<OP_DISTINTO, ~>)
                                                          30:32-->L, G.token(<NOT, ~>)
6:7--> If (pos = TS(lexema)
                                                         0:33-->L
        then G.token(< Id , pos>)
                                                          33:34-->L, G.token(<OP_MEN_IG, ~>)
       Else if (pos = T.palreservada(lexema))
                                                         33:35-->L, G.token(<OP_MEN, ~ >)
       then G.token(<PR, pos>)
                                                         0:36-->L
       else pos = Insertar Ts(lexema),
                                                          36:38-->L, G.token(<OP_MAY_IG, ~>)
            Gtoken(< Id, pos>)
                                                          36:37-->L, G.token(<OP_MAY, ~>)
0:8-->L
                                                          0:40-->L, G.token(<PARA, ~ >)
0:39-->G.token(<DIV, ~>)
                                                         0:41-->L, G.token(<PARC, ~>)
8:9-->L
                                                         0:42-->L, G.token(<LLAVA, ~ >)
9:9-->L
                                                         0:43-->L, G.token(<LLAVC, ~>)
9:10-->L
```

#### 3. Diseño Analizador Sintáctico

#### 3.1. Gramática

```
Terminales = { var id; if () while } { switch case break default int bool string return print
                         prompt function , || \&\& == != <> -= != +- */\% CAD : cteent ctebool
}
NoTerminales = { PBTS_SFHLQAKXCE_ER_RU_UV_VW_WZ_ZG_GD}
Axioma = P
Producciones = {
P -> B P
                                    L -> lambda
                                                                        _<mark>V</mark> -> < W _V
P -> F P
                                    Q -> , E Q
                                                                         _V -> lambda
P -> lambda
                                    Q -> lambda
                                                                         W -> Z _W
B -> var T id;
                                    A \rightarrow T id K
                                                                        _W -> + Z _W
B \rightarrow if(E)S
                                    A -> lambda
                                                                         _W -> - Z _W
B -> while (E) {C}
                                    K \rightarrow T id K
                                                                         W -> lambda
B -> switch ( E ) { D }
                                    K -> lambda
                                                                         Z -> G _Z
B -> S
                                    X -> E
                                                                        _Z -> * G _Z
                                                                         _Z -> / G _Z
T -> int
                                    X -> lambda
                                                                         _Z -> % G _Z
T -> bool
                                    C -> B C
T -> string
                                    C -> lambda
                                                                         Z -> lambda
S \rightarrow id S
                                    E \rightarrow R E
                                                                         G -> ! G
S -> return X;
                                    _E -> || R _E
                                                                         G \rightarrow id G
S -> print (E);
                                    E -> lambda
                                                                         G \rightarrow (E)
                                                                         G -> cteent
S -> prompt ( id ) ;
                                    R -> U _R
S -> break;
                                    _R -> && U _R
                                                                         G -> CAD
S \rightarrow = E;
                                                                         G -> ctebool
                                    _R -> lambda
S -> -= E;
                                    U -> V ∪
                                                                         G -> (L)
                                    _U -> == V _U
                                                                         _G -> lambda
_S -> (L);
F -> function H id ( A ) { C }
                                    _U -> != V _U
                                                                         D -> case cteent : C D
                                    _U -> lambda
H -> T
                                                                         D -> default : C
H -> lambda
                                    ∨ -> W _V
                                                                         }
L -> E Q
                                    _{\sf V} \rightarrow > {\sf W}_{\sf V}
```

#### 3.2. Demostración LL(1)

Puesto que nuestro método de análisis sintáctico es el "Descendente Recursivo" nuestra gramática debe ser una gramática LL(1). Este tipo de gramáticas son no ambiguas, no recursivas por la izquierda y factorizadas por la izquierda. A parte de estas condiciones necesarias, todas las producciones de la gramática con mismo antecedente deben cumplir (dos a dos) la condición LL(1):

- Si A → α<sub>1</sub>, A → α<sub>2</sub> ∈ G => First(α<sub>1</sub>) ∩ First(α<sub>2</sub>) = Ø Es decir, que no tengan nada en común.
- Si A  $\rightarrow \alpha_1$ , A  $\rightarrow \alpha_2 \in G$  y  $\lambda \in First(\alpha_1) => Follow(A) \cap First(\alpha_2) = \emptyset$ Pertenece a una de las reglas.

Tras eliminar las recursividades por la izquierda y factorizar nuestra gramática empleamos la herramienta SDGLL1 proporcionada en la web de la asignatura para comprobar que nuestra gramática es LL(1). Tras analizar la gramática en la herramienta obtuvimos los First y Follows de las distintas reglas, así como la confirmación de que la gramática es LL(1).

### 3.3. Analizador descendente recursivo.

A continuación, mostramos el código de los procedimientos que realizarán el análisis sintáctico:

```
Proc T;
                                                                                                                                   if(st \in First(int)={ int }) then
print("Des ");
                                                                                                                                                   print("9");
Proc P;
                                                                                                                                                   EqT(int);
                 if(st \in First(B \; P) = \{ \; break \; id \; if \; print \; prompt \; return \; switch \; var \; while \; \})
then
                                                                                                                                  else if(st ∈ First(bool)={ bool }) then
                                  print("1");
                                                                                                                                                   print("10");
                                                                                                                                                   EqT(bool);
                                  P();
                                                                                                                                  else if(st \epsilon First(string)={ string }) then
                 else if(st \in First(F P)={ function }) then
                                                                                                                                                   print("11");
                                  print("2");
                                                                                                                                                   EqT(string);
                                  P();
                                                                                                                                  else ERROR;
                                                                                                                 END;
                 else if(st ∈ Follow(P)={$})
                                  print("3");
                                                                                                                 Proc S;
                 else ERROR:
                                                                                                                                  if(st \in First(id S')={ id }) then
END;
                                                                                                                                                   print("12");
                                                                                                                                                   EqT(id);
                 if(st \in First(var T id ;)={ var }) then
                                                                                                                                                   S'();
                                  print("4");
                                                                                                                                  else if(st ε First(return X ;)={ return }) then
                                  EqT(var);
                                  T();
                                                                                                                                                   print("13");
                                  EqT(id);
                                                                                                                                                   EqT(return);
                                  EqT(;);
                                                                                                                                                   X();
                 else if(st \in First(if ( E ) S)={ if }) then
                                                                                                                                                   EqT(;);
                                                                                                                                  else if(st ∈ First(print ( E ) ;)={ print }) then
                                  print("5");
                                  EqT(if);
                                                                                                                                                   print("14");
EqT(print);
                                  EqT(();
                                  E():
                                                                                                                                                   EqT(();
                                  EqT());
                                                                                                                                                   EqT());
                                                                                                                                                   EqT(;);
                 else if(st \epsilon First(while ( E ) { C })={ while }) then
                                                                                                                                  else if(st \epsilon First(prompt ( id ) ;)={ prompt }) then
                                  print("6");
EqT(while);
                                                                                                                                                   print("15");
                                                                                                                                                   EqT(prompt);
                                  E();
                                 EqT());
EqT({);
                                                                                                                                                   EqT(();
                                                                                                                                                   EqT(id);
                                                                                                                                                   EqT());
                                  C();
                                                                                                                                                   EqT(;);
                                  EqT(});
                                                                                                                                   else if(st \epsilon First( break ;)={ break }) then
                 else if(st \in First(switch ( E ) { D })={ switch }) then
                                                                                                                                                   print("16");
                                  print("7");
                                                                                                                                                   EqT(break);
                                  EqT(switch);
                                                                                                                                                   EqT(;);
                                  EqT(();
                                                                                                                                   else ERROR:
                                  EqT());
                                                                                                                 END;
                                  EqT({);
                                  D();
                                  EqT(});
                 else if(st \in First(S)={ break id print prompt return }) then
                                  print("8");
                                  S();
                 else ERROR:
END:
```

```
Proc S':
                if(st \in First(= E ;)={ = }) then
                                                                                                            Proc A;
                                                                                                                             if(st \in First(T id K)={ int bool string }) then
                                print("17");
                                                                                                                                             print("27");
                                EqT(=);
                                E();
EqT(;);
                                                                                                                                             T();
EqT(id);
                                                                                                                                             K();
                else if(st ∈ First(-= E ;)={ -= }) then
                                                                                                                             else if(st \in Follow(A)={ ); }) then
                                print("18");
                                                                                                                                             print("28");
                                EqT(-=);
                                E();
                                EqT(;);
                                                                                                                             else ERROR;
                                                                                                            END;
                else if(st \epsilon First(( L ) ;)={ ( }) then
                                                                                                            Proc K:
                                                                                                                             if(st \in First(, T id K)={ , }) then
                                print("19");
                                EqT(();
                                                                                                                                             print("29");
                                EqT());
                                                                                                                                             EqT(,);
                                                                                                                                             T();
EqT(id);
                                EqT(;);
                else ERROR;
                                                                                                                                             K();
END;
                                                                                                                             else if(st \in Follow(K)={ ) }) then
Proc F;
                if(st \in First(function H id ( A ) { C })={ function }) then
                                                                                                                                             print("30");
                                print("20");
                                                                                                                             else ERROR;
                                EqT(function);
                                                                                                             END;
                                H();
EqT(id);
EqT(();
                                                                                                            Proc X;
                                                                                                                             if(st \in First(E)={ ! ( cte_bool CAD cte_ent id }) then
                                A();
                                EqT());
                                                                                                                                             print("31");
                                EqT({);
                                C();
EqT(});
                                                                                                                             else if(st \in Follow(X)={ ; }) then
                else ERROR;
                                                                                                                                             print("32");
END;
                                                                                                                             else ERROR;
                                                                                                            END:
Proc H:
                if(st \in First(T)={ int bool string }) then
                                                                                                            Proc C;
                                print("21");
                                                                                                                             if(st \in First(B \ C) = \{ \ break \ id \ if \ print \ prompt \ return \ switch \ var \ while \ \})
                                                                                                             then
                else if(st \in Follow(H)={ id }) then
                                                                                                                                             print("33");
                                                                                                                                             B():
                                print("22");
                                                                                                                                             C();
                                                                                                                             else if(st \in Follow(C)={ case } default }) then
                else ERROR;
END:
                                                                                                                                             print("34");
Proc L;
                if(st \in First(E Q)={ ! ( cte_bool CAD cte_ent id }) then
                                                                                                                             else ERROR;
                                                                                                            END:
                                print("23");
                                E():
                                                                                                            Proc E:
                                Q();
                                                                                                                             if(st \in First(R E')={ ! ( cte_bool CAD cte_ent id }) then
                else if(st \epsilon Follow(L)={ ) }) then
                                                                                                                                             print("35");
                                                                                                                                             R();
                                print("24");
                                                                                                                                             E'();
                else ERROR;
                                                                                                                             else ERROR;
END;
                                                                                                            END;
Proc Q;
                                                                                                            Proc E';
                if(st \in First(, E Q)={ , }) then
                                                                                                                             if(st \epsilon First(|| R E')={ || }) then
                                print("25");
                                                                                                                                             print("36");
                                EqT(,);
                                                                                                                                             EqT(||);
                                                                                                                                             R();
                                Q();
                                                                                                                                             E'();
                else if(st \in Follow(Q)={ ) }) then
                                                                                                                             else if(st \in Follow(E')={ ) , ; }) then
                                print("26");
                                                                                                                                             print("37");
                                                                                                                             else ERROR:
                else ERROR;
                                                                                                            END;
END;
```

```
Proc R;
                                                                                                        Proc W;
               if(st \in First(U R')={ ! ( cte_bool CAD cte_ent id }) then
                                                                                                                        if(st \in First(Z W')={ ! ( cte_bool CAD cte_ent id }) then
                               print("38");
                                                                                                                                        print("49");
                                                                                                                                        Z();
W'();
                               U();
                               R'();
               else ERROR;
                                                                                                                        else ERROR;
END;
                                                                                                        END;
Proc R':
                                                                                                        Proc W':
               if(st \in First(&& U R')={ && }) then
                                                                                                                        if(st \in First(+ Z W')={ + }) then
                               print("39");
                                                                                                                                        print("50");
                               EqT(&&);
                                                                                                                                        EqT(+);
                               U();
                                                                                                                                        Z():
                               R'();
                                                                                                                                        W'();
               else if(st \in Follow(R')={ ) , ; || }) then
                                                                                                                        else if(st ∈ First(- Z W')={ - }) then
                               print("40");
                                                                                                                                        print("51");
                                                                                                                                        EqT(-);
               else ERROR;
                                                                                                                                        Z();
END;
                                                                                                                                        W'();
                                                                                                                        else if(st \in Follow(W')={ != && ) , ; < == > | | }) then
Proc U;
               if(st \in First(V U')={ ! ( cte_bool CAD cte_ent id }) then
                                                                                                                                        print("52");
                               print("41");
                               V();
                                                                                                                        else ERROR;
                               U'();
                                                                                                        END;
               else ERROR;
                                                                                                        Proc Z:
END;
                                                                                                                        if(st \in First(G Z')={ ! ( cte_bool CAD cte_ent id }) then
Proc U';
                                                                                                                                        print("53");
               if(st \in First(== V U')={ == }) then
                                                                                                                                        G();
                                                                                                                                       Z'();
                               print("42");
                               EqT(==);
                                                                                                                        else ERROR;
                               V();
                                                                                                         END;
                               U'();
                                                                                                        Proc Z':
               else if(st \in First(!= V U')={ != }) then
                                                                                                                        if(st ∈ First(* G Z')={ * }) then
                               print("43");
                               EqT(!=);
                                                                                                                                        print("54");
                               V();
                                                                                                                                        EqT(*);
                               U'();
                                                                                                                                        G();
                                                                                                                                        Z'();
               else if(st \in Follow(U')={ && ) , ; || }) then
                                                                                                                        else if(st \epsilon First(/ G Z')={ / }) then
                               print("44");
                                                                                                                                        print("55");
               else ERROR;
                                                                                                                                        EqT(/);
END;
                                                                                                                                        G();
                                                                                                                                        Z'();
Proc V;
               if(st \in First(W V')={ ! ( cte_bool CAD cte_ent id }) then
                                                                                                                        else if(st ∈ First(% G Z')={ % }) then
                                                                                                                                        print("56");
                               print("45");
                               .
W();
                                                                                                                                        EqT(%);
                               V'();
                                                                                                                                        Z'();
               else ERROR:
END;
                                                                                                                        else if(st \in Follow(Z')={ != && ) + , - ; < == > | | }) then
Proc V';
                                                                                                                                        print("57");
               if(st \in First(> W V')={ > }) then
                                                                                                                        else ERROR;
                               print("46");
                                                                                                        END:
                               EqT(>);
                               W();
                               V'();
               else if(st \in First(< W V')={ < }) then
                               print("47");
                               EqT(<);
                               W();
                               V'();
               else if(st \in Follow(V')={ != && ) , ; == | | }) then
                               print("48");
               else ERROR;
END;
```

```
Proc G;
                 if(st \in First(! G)={ ! }) then
                                  print("58");
                                  EqT(!);
G();
                 else if(st ∈ First(id G')={ id }) then
                                  print("59");
EqT(id);
G'();
                 else if(st \in First(( E ))={ ( }) then
                                  print("60");
                                  EqT(();
                                  E();
EqT());
                 else if(st \epsilon First(cte_ent)={ cte_ent }) then
                                  print("61");
                                  EqT(cte_ent);
                 else if(st \epsilon First(CAD)={ CAD }) then
                                  print("62");
EqT(CAD);
                 else if(st \epsilon First(cte_bool)={ cte_bool }) then
                                  print("63");
                                  EqT(cte_bool);
                 else ERROR;
END;
Proc G';
                 if(st \in First(( L ))={ ( }) then
                                  print("64");
                                  EqT(();
                                  L();
EqT());
                 else if(st \in Follow(G')={ != % && ) * + , - / ; < == > | | }) then
                                  print("65");
                 else ERROR;
END;
Proc D;
                 if(st \in First(case cte_ent : C D)={ case }) then
                                  print("66");
                                  EqT(case);
EqT(cte_ent);
EqT(:);
                                  C();
                                  D();
                 else if(st \epsilon ) then
                                  print("67");
EqT(default);
                                  EqT(:);
                                  C();
                 else ERROR;
END;
```

#### 4. Diseño Analizador Semántico

A continuación, la traducción dirigida por la sintaxis en formato de esquema de traducción para el análisis semántico:

```
P -> {B.tipoSwitch=false} B P
P \rightarrow F P
                {}
P -> lambda
                {}
B -> var {declaración=true} T id {insertaTipoTS(id.lex, T.Tipo); insertaDespTS(id.pos, T.desp);}
{declaración=false}; {B.tipoRet='tipo_vacio' }
B -> if (E) { if(E.tipo!=bool) then ERROR #1 else S.switch = B.switch } S
{ B.tipoRet = S.TipoRet; }
B -> while (E) { if(E.tipo!=bool) then ERROR #2 elseC.switch = B.switch } { C }
        {B.tipoRet=C.tipoRet}
B -> switch (E) { if(E.tipo!=int) ERROR #3 } { D } {B.tipoRet=D.tipoRet}
B -> {S.switch = B.switch} S {B.tipoRet=S.tipoRet}
T -> int
                {T.Tipo = int}
T -> bool
                {T.Tipo = bool}
T -> string
                {T.Tipo = string}
S -> id _S { if(buscaTipo(id.pos) j = _S.tipo->t ERROR #4 S.tipoRet=tipo_vacio;
                   else if(buscaTipo(id.pos) == _S.tipo then ERROR #5; S.tipoRet=tipo_vacio;
                        else if(buscaTipo(id.pos) == tipo_vacio then insertaTipoTS(id.pos, int);
                        if(_S.tipo!=id.tipo) then ERROR #6 ;S.tipoRet=tipo_vacio; }
                                else Error}
S --> return X {if TSactual != then ERROR #7 TSL then S.tipoRet = X.tipo};
S -> print (E) {S.Tipo= if(E.tipo IN {int,bool,string} S.tipoRet=tipo_vacio; else ERROR #8 };
(E puede ser de cualquiera de los 3 tipos )
S -> prompt (id) {S.Tipo= if(buscaTipo(id.pos) IN {int, string} then S.tipoRet=tipo_vacio; else
ERROR #9)};
(id tiene que ser entero o cadena)
```

```
S -> break; {if(S.switch = false then ERROR #10 ("Break fuera de switch")
                 else S.tipoRet=tipo_vacio;}
_S -> = E ; {_S.tipo = E.tipo}
_S -> -= E; {_S.tipo = if(E.tipo==int) then int else ERROR #11 }
_S -> ( L ); {_S.tipo=L.tipo}
F -> function {declaración=true} H id {TL=creaTSL(); desp=0;} ( A ) {declaración=false}
{insertaTipoTS(id.pos, A.Tipo->H.Tipo); declaración=false} { C {if(C.tipoRet != H.tipo) then
ERROR #12; destruirTSL()} }
H -> T {H.Tipo = T.Tipo}
H -> lambda {H.Tipo = tipo_vacio}
L -> E Q {L.Tipo = E.tipo x Q.Tipo}
L -> lambda {L.tipo = tipo_vacio}
Q \rightarrow , E Q \{Q.Tipo = E.tipo x Q_1.Tipo\}
Q -> lambda {Q.Tipo = tipo_vacio}
A -> T id {insertatipoTS(id.pos,T.tipo); insertaDespTS(id.pos,desp); desp=desp++} K {A.Tipo =
T.tipo x K.tipo}
A -> lambda {A.Tipo = tipo_vacio}
K -> , T id {insertatipoTS(id.pos,T.tipo); insertaDespTS(id.pos,desp); desp=desp++} K<sub>1</sub> {K.Tipo =
T.tipo x K<sub>1</sub>.tipo}
K -> lambda
                 {K.Tipo = tipo_vacio}
X -> E
                 {X.Tipo = E.Tipo}
X -> lambda
               {X.Tipo = tipo_vacio}
C -> {B.switch = C.switch; C<sub>1</sub>.switch = B.switch} B C<sub>1</sub>
{ C.TipoRet = if(B.tipoRet == C<sub>1</sub>. tipoRet) then B.tipoRet
        Else if(B.tipoRet == tipo_vacio) then C<sub>1</sub>.tipoRet
                 Else if(C<sub>1</sub>.tipoRet == tipo_vacio) then B.tipoRet
                         Else Error #13 }
C -> lambda
                 {C.TipoRet = tipo_vacio}
```

```
{E.Tipo = if(_E.Tipo== tipo_vacio) then R.Tipo else
E -> R _E
                               if(_E.Tipo== bool AND R.Tipo == bool) then bool else
                                       ERROR #14 }
_E -> | | R _E {_E.Tipo = if( (R.Tipo==bool) AND (_E1.Tipo==bool OR tipo_vacio) )then bool else
                       ERROR #14}
_E -> lambda
                       {E.Tipo = tipo_vacio }
E puede ser: int, string, bool, Error
_E puede ser: bool, vacío, Error
R -> U R
               {R.Tipo = if(_R.Tipo== tipo_vacio) then U.Tipo else
                       if(_R.Tipo== bool AND U.Tipo == bool) then bool else ERROR #14}
_R -> && U _R {_R.Tipo = if( (U.Tipo==bool) AND (_R1.Tipo==bool OR tipo_vacio) )then bool else
                       ERROR #14}
_R -> lambda
                       {R.Tipo = tipo_vacio }
R puede ser: int, string, bool, Error
_R puede ser: bool, vacío, Error
U -> V _U
                      {U.Tipo = if(_U.Tipo== tipo_vacio) then V.Tipo else
                               if(_U.Tipo== bool AND V.Tipo == int) then bool else
                                       ERROR #14}
_U \rightarrow == V _U \{_U.Tipo=if((V.Tipo==int))\} AND (_U_1.Tipo==tipo_vacio)) then bool else ERROR #14}
_U -> != V _U {_U.Tipo=if( (V.Tipo==int) AND (_U<sub>1</sub>.Tipo==tipo_vacio) )then bool else ERROR #14}
_U -> lambda
                       {_U.Tipo = tipo_vacio }
U puede ser: int, string, bool, Error
_U puede ser: bool, vacío, Error
V -> W _V
              {V.Tipo = if(_V.Tipo== tipo_vacio) then W.Tipo else
```

```
_V -> > W _V {_V.Tipo=if( (W.Tipo==int) AND (_V<sub>1</sub>.Tipo==tipo_vacio) )then bool else
                        ERROR #14}
_V -> < W _V {_V.Tipo=if( (W.Tipo==int) AND (_V<sub>1</sub>.Tipo==tipo_vacio) )then bool else
                        ERROR #14}
_V -> lambda
                       {_V.Tipo = tipo_vacio }
V puede ser: int, string, bool, Error
_V puede ser: bool, vacío, Error
W \rightarrow Z W
                   {W.Tipo = if(_W.Tipo== tipo_vacio) then Z.Tipo else
                                if(_W.Tipo== int AND Z.Tipo == int) then int else
                                        ERROR #14}
_W -> + Z _W {_W.Tipo = if( (Z.Tipo==int) AND (_W<sub>1</sub>.Tipo==int OR tipo_vacio) )then int else
                                ERROR #14}
_W -> - Z _W {_W.Tipo = if( (Z.Tipo==int) AND (_W<sub>1</sub>.Tipo==int OR tipo_vacio) )then int else
ERROR #14}
W -> lambda
                       {_W.Tipo = tipo_vacio }
W puede ser: int, string, bool, Error
_W puede ser: int, vacío, Error
Z -> G _Z
                       {Z.Tipo = if(_Z.Tipo== tipo_vacio) then G.Tipo else
                                if(_Z.Tipo== int AND G.Tipo == int) then int else
                                        ERROR #14 }
```

if(\_V.Tipo== bool AND W.Tipo == int) then bool else

**ERROR #14** }

```
_Z -> * G _Z {_Z.Tipo = if( (G.Tipo==int) AND (_Z<sub>1</sub>.Tipo==int OR tipo_vacio) ) then int else
                                 ERROR #14}
_Z -> / G _Z {_Z.Tipo = if( (G.Tipo==int) AND (_Z<sub>1</sub>.Tipo==int OR tipo_vacio) ) then int else
                                 ERROR #14}
_Z -> % G _Z<sub>1</sub> {_Z.Tipo = if( (G.Tipo==int) AND (_Z<sub>1</sub>.Tipo==int OR tipo_vacio) ) then int else
                                 ERROR #14}
_Z -> lambda
                        {_Z.Tipo = tipo_vacio }
Z puede ser: int, string, bool, Error
_Z puede ser: int, tipo_vacio, Error
                {G.Tipo = if(G<sub>1</sub>.Tipo==bool) then bool else ERROR #15}
G-> !G<sub>1</sub>
G -> id _G
                         {G.Tipo = if(_G.Tipo== tipo_vacio) then buscaTipoTS(id.pos) else
                                 If (buscaTipoTS(id.pos)==_G.Tipo->t then t else ERROR #14)
G->(E)
                         {G.Tipo = E.Tipo}
G -> cteent
                         {G.Tipo = int}
G -> CAD
                         {G.Tipo = string}
G -> ctebool
                         {G.Tipo = bool}
_G -> ( L )
                         {_G.Tipo = L.Tipo}
                        {_G.Tipo = tipo_vacio }
_G -> lambda
G puede ser: int, string, bool, Error
_G puede ser: tipo_vacio, arg1 x arg2 x ... x argN
D -> case cteent : {C.switch = true} C D<sub>1</sub>
{D.tipoRet=if(C.tipoRet == D1.tipoRet) then C.tipoRet
        Else if(C.tipoRet == tipo_vacio) then D1.tipoRet
                Else if(D1.tipoRet == tipo_vacio) then C.tipoRet
                         Else Error
D -> default : {C.switch = true } C {D.tipoRet = C.tipoRet}
}
```

#### 5. Tabla de símbolos

La tabla de símbolos consta de dos tablas, una general y otra local. La tabla general almacena los identificadores globales tanto de variables como de funciones. Los atributos que almacena de una variable son: tipo ('int','bool','string') y el desplazamiento que refleja el tamaño reservado para el valor almacenado en dicha variable dependiendo del tipo de la variable. Los atributos que almacena para las funciones son: parámetros de entrada (de los cuales almacena el tipo y el modo de paso de parámetros que va a ser siempre 1 que implica paso por valor), etiqueta de la función, tipo de retorno y numero de parámetros. Cada tabla local se crea al declarar una función. Almacena los argumentos de entrada, el tipo de retorno y las variables locales declaradas dentro de la función. Una vez terminada la declaración de la función ésta se destruye y vuelca su información en el fichero de tabla de símbolos.

La tabla incluye un flag de declaración que modifica el comportamiento de la inserción de identificadores en la tabla de símbolos. Si el flag está activado se inserta el valor en la tabla de símbolos actual si no se ha hecho previamente. Si está desactivado pero se encuentra un identificador no declarado previamente, éste se almacena en la tabla global con tipo 'int'.

#### 6. Gestor de errores

En este apartado se muestran los distintos mensajes de error que puede mostrar el gestor de errores. Nuestro compilador, en el momento en que detecta un fallo, detiene el análisis, muestra por pantalla el mensaje de error correspondiente y vuelca la información en los ficheros de salida.

#### **Errores Léxicos:**

Error #1: La cte entera supera el tamaño máximo permitido.

Error #2: carácter no permitido.

#### **Errores Sintácticos:**

Error #1: sintaxis incorrecta.

Error #2: El tipo de retorno de una función debe ser un tipo básico o vacío.

Error #3: Error en llamada de función.

Error #4: Parámetros formales de declaración de función incorrectos.

Error #5: Expresión mal construida.

Error #6: sentencia case/default incorrecta.

Error #7: token recibido distinto al esperado.

#### **Errores Semánticos:**

Error #1: La condición del 'if' debe ser de tipo booleano.

Error #2: La condición del 'while' debe ser de tipo booleano.

Error #3: La condición del 'switch' debe ser de tipo entero.

Error #4: Llamada a función con parámetro incorrectos.

Error #5: La variable no es de tipo función y por lo tanto no puede hacerse una llamada.

Error #6: Tipo erróneo en asignación.

Error #7: Sentencia return fuera de función.

Error #8: print() debe llamarse con un argumento de tipo entero, booleano o string.

Error #9: prompt() debe llamarse con una variable de tipo entero o string como argumento.

Error #10: break fuera de sentencia switch.

Error #11: la sentencia de asignación con resta solo admite operandos enteros

Error #12: el tipo de retorno de la función declarado: '+hTipo+' no coincide con el tipo devuelto.

Error #13: sentencias de retorno en función inconsistentes.

Error #14: expresión incorrecta.

#### Anexo

### **Pruebas Correctas:**

#### Prueba 1:

```
*/
//pepe
function string pideTexto ()
{
      print ('Introduce un texto');
      prompt (texto);
 return 'cad';
}
texto = pideTexto();
function imprime (string msg, int p, int p1)
{
      print (msg);
pideTexto();
var string textoAux;
textoAux = textoAux;
//imprime (textoAux);
```

# Tokens

<pr,9></pr,9>	<id,12></id,12>
<pr,3></pr,3>	<para,></para,>
<id,10></id,10>	<pr,3></pr,3>
<fin,></fin,>	<id,20></id,20>
<pr,8></pr,8>	<sig,></sig,>
<pr,3></pr,3>	<pr,1></pr,1>
<id,11></id,11>	<id,21></id,21>
<para,></para,>	<sig,></sig,>
<parc,></parc,>	<pr,1></pr,1>
<llava,></llava,>	<id,22></id,22>
<pr,12></pr,12>	<parc,></parc,>
<para,></para,>	<llava,< td=""></llava,<>
<cad,"introduce texto"="" un=""></cad,"introduce>	<pr,12></pr,12>
<parc,></parc,>	<para,></para,>
<fin,></fin,>	<id,20></id,20>
<pr,13></pr,13>	<parc,></parc,>
<para,></para,>	<fin,></fin,>
<id,10></id,10>	<llavc,< td=""></llavc,<>
<parc,></parc,>	<id,11></id,11>
<fin,></fin,>	<para,></para,>
<pr,7></pr,7>	<parc,></parc,>
<cad,"cad"></cad,"cad">	<fin,></fin,>
<fin,></fin,>	<pr,9></pr,9>
<llavc,></llavc,>	<pr,3></pr,3>
<id,10></id,10>	<id,13></id,13>
<asig,></asig,>	<fin,></fin,>
<id,11></id,11>	<id,13></id,13>
<para,></para,>	<asig,></asig,>
<parc,></parc,>	<id,13></id,13>
<fin,></fin,>	<fin,></fin,>
<pr,8></pr,8>	
•	

### Árbol

```
"p(1)
"8(8)
"5(12)
id
"5(17)
"6(35"
Gramática: C:\Users\PABLO\Desktop\Arbol1.txt
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    Parse: C:\Users\PABLO\Proyectos\PDLG64\src\Salida\parse.txt
P(1)
B(4)
         var
T (11)
             string
     F (20)
function
H (21)
'Y (11)
string
id
(
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   (
*A (28)
*lambda
         _a)
lambda
]
{
    C (33)
    *B (8)
    *S (14)
    print
    (
    *E (35)
    *R (38)
    *V (41)
    *V (45)
    *V (40)
    *Z (53)
    *G (62)
    *CAD
    *Z (57)
    *lambda
    *W (52)
    lambd>
    V //
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                ;
"P (2)
"F (20)
"function
"H (22)
"lambda
                                                                   "_W (52)
Tambda
"_V (48)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               A (27)
T (11)
string
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       id
K (29)
                                                       "_U (44)
"Tambda
"_R (40)
"Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         T (9)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           id
K (29)
                                                               _E (37)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               Ť (9)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  int
id
K (30)
Tambda
                                  C (33)
B (8)
S (15)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    rambda

| Cass | Sass |
                                        C (33)
B (8)
S (13)
                                                       For (33)

return

X (31)

E (35)

R (38)

U (41)

V (45)

V (45)

G (52)

CAD

Z (57)

Tambda

U (48)

Tambda

U (48)

Tambda

U (48)

Tambda

E (37)

Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       C (34)
                                                C (34)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         "P (1)
"B (8)
"S (12)
| id
| _S (19)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 (
L (24)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 P (1)
B (4)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       var
T (11)
```

# Tabla de Símbolos

TABLA PRINCIPAL #1:	ATRIBUTOS:		
* LEXEMA: 'texto'	+despl: 130		
ATRIBUTOS:	+tipo: 'string'		
+despl: 0			
+tipo: 'string'			
	TSL de la funcion 'pideTexto' #2:		
* LEXEMA: 'pideTexto'			
ATRIBUTOS:	TSL de la funcion 'imprime' #3:		
+numParam: 0	* LEXEMA: 'msg' (parametro de		
+TipoRetorno: 'string'	entrada de la funcion)		
+EtiqFuncion:	ATRIBUTOS:		
'ETIpideTexto2'	+despl: 0		
	+tipo: 'string'		
* LEXEMA: 'imprime'			
ATRIBUTOS:	* LEXEMA: 'p' (parametro de		
+numParam: 3	entrada de la funcion)		
+TipoRetorno: 'tipo_vacio'	ATRIBUTOS:		
+EtiqFuncion: 'ETIimprime3'	+despl: 128		
+TipoParam1: 'string'	+tipo: 'int'		
+ModoParam1: 1			
+TipoParam2: 'int'	* LEXEMA: 'p1' (parametro de		
+ModoParam2: 1	entrada de la funcion)		
+TipoParam3: 'int'	ATRIBUTOS:		
+ModoParam3: 1	+despl: 130		
	+tipo: 'int'		
* LEXEMA: 'textoAux'			

# Prueba 2:

```
var int a; var bool b;
b = false && 2 == 2%2 || true && a > 34 + 3*a;
var bool c;
c = a > a;
if (c) a -= 1;
if (2 < a) a -= 4;
a = a + 4;
print (a);
print (b);</pre>
```

#### Tokens

<PR,9> <ID,10> <PR,1> <FIN,> <PR,4> <ID,10> <FIN,> <PARA,> <PR,9> <ID,12> <PR,2> <PARC,> <ID,11> <ID,10> <FIN,> <ASIGR,> <ID,11> <cteent,1> <ASIG,> <FIN,> <ctebool,0> <PR,4> <AND,> <PARA,> <cteent,2> <cteent,2> <OPIG,> <OPMEN,> <ID,10> <cteent,2> <MOD,> <PARC,> <cteent,2> <ID,10> <OR,> <ASIGR,> <ctebool,1> <cteent,4> <AND,> <FIN,> <ID,10> <ID,10> <OPMAY,> <ASIG,> <cteent,34> <ID,10> <MAS,> <MAS,> <cteent,3> <cteent,4> <MUL,> <FIN,> <ID,10> <PR,12> <FIN,> <PARA,> <PR,9> <ID,10> <PR,2> <PARC,> <ID,12> <FIN,> <FIN,> <PR,12> <ID,12> <PARA,> <ASIG,> <ID,11> <ID,10> <PARC,> <OPMAY,> <FIN,>

### Árbol

```
Arbol resultado de:
Iramática: C:\Users\PABLO\Desktop\Arbol1.txt
                                                                                                                                                                                                                                             var
*T (10)
Parse: C:\Users\PABLO\Proyectos\PDLG64\src\Salida\parse.txt
 P (1)
B (4)
                                                                                                                                                                                                                                           P (1)
B (8)
S (12)
    var
T (9)
int
id
                                                                                                                                                                                                                                                A

4

5 (17)

E (35)

R (38)

U (41)

V (45)

W (40)

2 (53)

G (59)

id

G 7
   P (1)
B (4)
       var
T (10)
bool
id
     "P (1)
"B (8)
"S (12)
id
"S (17)
"E
                                                                                                                                                                                                                                                                      id
_G (65)
             "_S (17)
"E (35)
"R (38)
"U (41)
"V (45)
"W (49)
"Z (53)
"G (63)
"ctebool
"Z (57)
"lambde
"_W (52)
"tambde
"_V
                                                                                                                                                                                                                                                                  _G (65)
Tambda
_Z (57)
Tambda
_W (52)
Tambda
                                                                                                                                                                                                                                                                 "_V (46)
                                                                                                                                                                                                                                                                  "W (49)
"2 (53)
"G (59)
Id
"G (65)
Tambda
"Z (57)
Tambda
                            "_W (52)
lambda
                                                                                                                                                                                                                                                                    Tambda
W (52)
                        "_V (48)
"lambda
"_U (44)
"lambda
                                                                                                                                                                                                                                                                  Tambda
"_V (48)
Tambda
                     amb.
_R (39)
88
*U (41)
*V (45)
*W (49)
*Z (53)
*G (61)
*Cleant
_Z (*
                                                                                                                                                                                                                                                       "Lu (44)
"Lambda
"_R (40)
"Lambda
"_E (37)
"Lambda
                                                                                                                                                                                                                                             P (1)
B (5)
                                                                                                                                                                                                                                                if
(
"E (35)
"R (38)
"U (41)
"Y (45)
"W (40)
"Z (53)
"G (50)
lid
"G
                                  Tambda
"_W (52)
                                _V (48)
                                   Tambda
                             "_U (42)
"==
"V (45)
"W (49)
"Z (53)
"G (61)
                                                                                                                                                                                                                                                        "G (50)
iid
"G (65)
Tambda
"Z (57)
Tambda
"W (52)
Tambda
"V (48)
Tambda
"U (44)
Tambda
                                      cteent
Z (56)
                                        %
G (61)
                                         cteent
"_Z (57)
lambda
                                                                                                                                                                                                                                                     lambda
_R (40)
lambda
_E (37)
lambda
                                     "_W (52)
                                         Tambda
                              "_V (48)
"lambda
"_U (44)
                                                                                                                                                                                                                                                  "S (12)
id
"S (18)
"E (35)
"R (38)
"U (41)
"V (45)
"W (40)
"Z (53)
"G (61)
"Cteent
"Z (57)
lambda
"W (5)"
                        lambda
R (40)
                   (am
_E (36)
| 11
| R (38)
| U (41)
| V (45)
| W (40)
| Z (53)
| G (63)
| Cubber
| _Z /r
                              Tambda
                  "_E (36)
                               "_Z (57)
lambda
"_W (52)
lambda
"_V (48)
                                                                                                                                                                                                                                                               Tambda

"_W (52)

Tambda

"_V (48)

Tambda

"_U (44)
                                                                                                                                                                                                                                                           lambda

"_R (40)

lambda

"_E (37)

lambda
                             R (39)
                        "_R (39)
8&
"U (41)
"V (45)
"W (49)
"Z (53)
"G (59)
Id
"_G (65)
"lambda
"_Z (57)
"lambda
                                                                                                                                                                                                                                                 *P (1)
*B (5)
if
                                                                                                                                                                                                                                                      °E (35)
```

### Tabla de Símbolos

# TABLA PRINCIPAL #1: \* LEXEMA: 'a' ATRIBUTOS: +despl: 0 +tipo: 'int' ----- \* LEXEMA: 'b' ATRIBUTOS: +despl: 2 +tipo: 'bool' ----- \* LEXEMA: 'c' ATRIBUTOS: +despl: 6 +tipo: 'bool'

## Prueba 3:

```
var int a;
var int b;
var int c;
print ( 'Introduce el primer operando' );
prompt (a);
print ( 'Introduce el segundo operando' );
prompt (b);
function int divide (int num1, int num2)
{
     return num1/num2;
}
c = divide (a, b);
print (c);
```

# Tokens

<pr,9></pr,9>	<id,13></id,13>
<pr,1></pr,1>	<para,></para,>
<id,10></id,10>	<pr,1></pr,1>
<fin,></fin,>	<id,20></id,20>
<pr,9></pr,9>	<sig,></sig,>
<pr,1></pr,1>	<pr,1></pr,1>
<id,11></id,11>	<id,21></id,21>
<fin,></fin,>	<parc,></parc,>
<pr,9></pr,9>	<llava,></llava,>
<pr,1></pr,1>	<pr,7></pr,7>
<id,12></id,12>	<id,20></id,20>
<fin,></fin,>	<div,></div,>
<pr,12></pr,12>	<id,21></id,21>
<para,></para,>	<fin,></fin,>
<cad,"introduce el="" primer<="" td=""><td><llavc,></llavc,></td></cad,"introduce>	<llavc,></llavc,>
operando">	<id,12></id,12>
<parc,></parc,>	<asig,></asig,>
<fin,></fin,>	<id,13></id,13>
<pr,13></pr,13>	<para,></para,>
<para,></para,>	<id,10></id,10>
<id,10></id,10>	<sig,></sig,>
<parc,></parc,>	<id,11></id,11>
<fin,></fin,>	<parc,></parc,>
<pr,12></pr,12>	<fin,></fin,>
<para,></para,>	<pr,12></pr,12>
<cad,"introduce el="" segundo<="" td=""><td><para,></para,></td></cad,"introduce>	<para,></para,>
operando">	<id,12></id,12>
<parc,></parc,>	<parc,></parc,>
<fin,></fin,>	<fin,></fin,>
<pr,13></pr,13>	
<para,></para,>	
<id,11></id,11>	
<parc,></parc,>	
<fin,></fin,>	
<pr,8></pr,8>	
<pr,1></pr,1>	

#### Árbol

#### Árbol resultado de:

Gramática: C:\Users\PABLO\Desktop\Arbol1.txt

```
Parse: C:\Users\PABLO\Proyectos\PDLG64\src\Salida\parse.txt
       var
T (9)
       int
id
     P (1)
B (4)
          var
T (9)
int
id
          P (1)
B (4)
               var
T (9)
int
id
             "$ (1)

"$ (8)

"$ (14)

print
(
"$ (35)

"R (38)

"R (38)

"U (41)

"V (45)

"W (40)

"Z (53)

"G (52)

"CAD

" Z (57)

"Tambda

" W (52)

Tambda
" U (44)

Tambda
" U (44)

Tambda
" L (40)

Tambda
" E (37)

Tambda
" L (40)

Tambda
                      P (1)
*8 (8)
"S (15)
                                prompt
(
id
)
                       "P (1)
"B (8)
"S (14)
print
(
"E (35)
"R (38)
                                        *R (38)
*U (41)
*V (45)
*V (49)
*Y (45)
*W (49)
*Z (53)
*G (62)
*CAD
*Lambda
*" Y (52)
*Tambda
*" Y (48)
*Tambda
*" U (44)
*Tambda
                           "P (1)
*8 (8)
"S (15)
prompt
(
id
)
                                      F (2)
F (20)
function
H (21)
T (9)
int
id
                                        (
*A (27)
*T (9)
```

```
*E (35)
*R (38)
*U (41)
*V (45)
*W (40)
*Z (53)
*G (50)
*Id
**G F
                                    A)

4

G (64)

(*L (23)

*E (35)

*R (38)

*U (41)

*V (45)

*W (40)

*Z (53)

*G (50)

iid

**G (7)
                                                             "G (50)
id
"G (65)
Tambda
"Z (57)
Tambda
"W (52)
Tambda
"448)
                                                         "_V (48)
"lambda
"_U (44)
                                                 lambda
"_R (40)
lambda
"_E (37)
lambda
                                              "Q (25)
"E (35)
"R (38)
"U (41)
"V (45)
"W (49)
"Z (53)
"G (59)
id
"_G (65)
"lambda
"_Z (57)
"lambda
"_W (52)
"lambda
"_V (48)
"lambda
"_U (44)
"lambda
                                                       "_U (44)
"lambda
"_R (40)
"lambda
                                                  "_E (37)
lambda
                                                 "Q (26)
Tambda
                          "_Z (57)
lambda
"_W (52)
lambda
"_V (48)
                               Tambda
                      "_U (44)
                   Tambda
"_R (40)
Tambda
               "_E (37)
lambda
"P (1)
*B (8)
"S (14)
print
    (14)
print
(*E (35)
*R (38)
*U (41)
*V (45)
*W (40)
*Z (53)
*G (50)
id
*"_G (65)
inambda
*_Z (57)
iam²
                   id

"G (65)

Tambda

"Z (57)

Tambda

"W (52)

Tambda

"V (48)

Tambda

"U (44)

Tambda
                   Tambda
"_R (40)
                         lambda
                "_E (37)
Tambda
      P (3)
          Tambda
```

#### Tabla de Símbolos

```
TABLA PRINCIPAL #1:
* LEXEMA: 'a'
      ATRIBUTOS:
      +despl: 0
      +tipo: 'int'
* LEXEMA: 'b'
      ATRIBUTOS:
      +despl: 2
      +tipo: 'int'
* LEXEMA: 'c'
      ATRIBUTOS:
      +despl: 4
      +tipo: 'int'
      _____
* LEXEMA: 'divide'
      ATRIBUTOS:
      +numParam: 2
      +TipoRetorno: 'int'
      +EtiqFuncion: 'ETIdivide2'
      +TipoParam1: 'int'
      +ModoParam1: 1
      +TipoParam2: 'int'
      +ModoParam2: 1
TSL de la funcion 'divide' #2:
* LEXEMA: 'num1' (parametro de entrada de la funcion)
      ATRIBUTOS:
      +despl: 0
      +tipo: 'int'
* LEXEMA: 'num2' (parametro de entrada de la funcion)
      ATRIBUTOS:
      +despl: 2
      +tipo: 'int'
```

#### Prueba 4:

```
var bool boolean;
function bool bisiesto (int a)
{
      return (a + 4 > 0 && a + 100 != 0 && a % 400 != 0);
function int dias (int m, int a)
      var int dd;
      print ('di cuantos dias tiene el mes');
      print (m);
      prompt(dd);
      if (bisiesto(a)) dd = dd % 1;
      return dd;
}
function bool esFechaCorrecta (int d, int m, int a)
{
      return m > 1 || m > 12 || d == 1 && d < dias (m, a);
function demo ()
{
      if (esFechaCorrecta(25, 10, 2018)) print ('OK');
 if (true) print ('OK');
}
var string A_A_A_;
demo();
```

# Tokens

<pr,9></pr,9>	<fin,></fin,>	<parc,></parc,>	<llava,></llava,>	<para,></para,>
<pr,2></pr,2>	<llavc,></llavc,>	<fin,></fin,>	<pr,7></pr,7>	<cteent,25></cteent,25>
<id,10></id,10>	<pr,8></pr,8>	<pr,4></pr,4>	<id,21></id,21>	<sig,></sig,>
<fin,></fin,>	<pr,1></pr,1>	<para,></para,>	<opmay,></opmay,>	<cteent,10></cteent,10>
<pr,8></pr,8>	<id,12></id,12>	<id,11></id,11>	<cteent,1></cteent,1>	<sig,></sig,>
<pr,2></pr,2>	<para,></para,>	<para,></para,>	<or,></or,>	<cteent,2018></cteent,2018>
<id,11></id,11>	<pr,1></pr,1>	<id,21></id,21>	<id,21></id,21>	<parc,></parc,>
<para,></para,>	<id,20></id,20>	<parc,></parc,>	<opmay,></opmay,>	<parc,></parc,>
<pr,1></pr,1>	<sig,></sig,>	<parc,></parc,>	<cteent,12></cteent,12>	<pr,12></pr,12>
<id,20></id,20>	<pr,1></pr,1>	<id,22></id,22>	<or,></or,>	<para,></para,>
<parc,></parc,>	<id,21></id,21>	<asig,></asig,>	<id,20></id,20>	<cad,"ok"></cad,"ok">
<llava,></llava,>	<parc,></parc,>	<id,22></id,22>	<opig,></opig,>	<parc,></parc,>
<pr,7></pr,7>	<llava,></llava,>	<mod,></mod,>	<cteent,1></cteent,1>	<fin,></fin,>
<para,></para,>	<pr,9></pr,9>	<cteent,1></cteent,1>	<and,></and,>	<pr,4></pr,4>
<id,20></id,20>	<pr,1></pr,1>	<fin,></fin,>	<id,20></id,20>	<para,></para,>
<mas,></mas,>	<id,22></id,22>	<pr,7></pr,7>	<opmen,></opmen,>	<ctebool,1></ctebool,1>
<cteent,4></cteent,4>	<fin,></fin,>	<id,22></id,22>	<id,12></id,12>	<parc,></parc,>
<opmay,></opmay,>	<pr,12></pr,12>	<fin,></fin,>	<para,></para,>	<pr,12></pr,12>
<cteent,0></cteent,0>	<para,></para,>	<llavc,></llavc,>	<id,21></id,21>	<para,></para,>
<and,></and,>	<cad,"di< td=""><td><pr,8></pr,8></td><td><sig,></sig,></td><td><cad,"ok"></cad,"ok"></td></cad,"di<>	<pr,8></pr,8>	<sig,></sig,>	<cad,"ok"></cad,"ok">
<id,20></id,20>	cuantos dias	<pr,2></pr,2>	<id,22></id,22>	<parc,></parc,>
<mas,></mas,>	tiene el mes ">	<id,13></id,13>	<parc,></parc,>	<fin,></fin,>
<cteent,100></cteent,100>	<parc,></parc,>	<para,></para,>	<fin,></fin,>	<llavc,></llavc,>
<opdistinto,></opdistinto,>	<fin,></fin,>	<pr,1></pr,1>	<llavc,></llavc,>	<pr,9></pr,9>
<cteent,0></cteent,0>	<pr,12></pr,12>	<id,20></id,20>	<pr,8></pr,8>	<pr,3></pr,3>
<and,></and,>	<para,></para,>	<sig,></sig,>	<id,14></id,14>	<id,15></id,15>
<id,20></id,20>	<id,20></id,20>	<pr,1></pr,1>	<para,></para,>	<fin,></fin,>
<mod,></mod,>	<parc,></parc,>	<id,21></id,21>	<parc,></parc,>	<id,14></id,14>
<cteent,400></cteent,400>	<fin,></fin,>	<sig,></sig,>	<llava,></llava,>	<para,></para,>
<opdistinto,></opdistinto,>	<pr,13></pr,13>	<pr,1></pr,1>	<pr,4></pr,4>	<parc,></parc,>
<cteent,0></cteent,0>	<para,></para,>	<id,22></id,22>	<para,></para,>	<fin,></fin,>
<parc,></parc,>	<id,22></id,22>	<parc,></parc,>	<id,13></id,13>	

### Árbol

```
Árbol resultado de:
Gramática: C:\Users\PABLO\Desktop\Arbol1.txt
  Parse: C:\Users\PABLO\Proyectos\PDLG64\src\Salida\parse.txt
                 var
T (10)
                   bool
id
               "P (2)
"F (20)
"function
"H (21)
"T (10)
"bool
id
                        id
(
A (27)
T (9)
int
id
"K (30)
Tambda
)
        (53)

*G (50)

(*E (35)

*R (38)

*U (41)

*V (45)

*W (49)

*2 (53)

*G (59)

lid

"G (59)

lid

"G (50)

Tambda

"2 (57)

Tambda

W (50)

*#

**Implementation of the content of the con
                                                                                                                                                                            +
Z (53)
G (61)
cteent
_Z (57)
lambda
                                                                                                                                                                            Tambda
"_W (52)
Tambda
                                                                                                                                                                        _V (46)
                                                                                                                                                                   "_V (49)
"Z (53)
"G (61)
"cteent
"_Z (57)
"lambda
"_W (52)
"lambda
"_V (48)
"_I (44)
                                                                                                                                   U (44)
lambda
R (39)
88
U (41)
V (45)
W (40)
C (59)
Id
                                                                                                                                                                                           id
"_G (65)
                                                                                                                                                                                "_Z (57)
lambda
"_W (50)
                                                                                                                                                                                       +
Z (53)
G (61)
                                                                                                                                                                                     cteent
"_Z (57)
Tambda
"_W (52)
Tambda
                                                                                                                                                                        "_V (48)
                                                                                                                                                                            lambda
_U (43)
                                                                                                                                                                   V (45)
W (49)
Z (53)
G (61)
cteent
```

```
"C (34)
Tambda
F (20)
Function
H (21)
Tr (9)
int
id
    (
A (27)
T (9)
int
        id
'K (29)
         Ť (9)
         int
id
K (30)
lambda
    C (33)
B (4)
         int
id
       "C (33)
8 (8)
"S (14)
print
            Tambda
"_U (44)
Tambda
"_R (40)
Tambda
"_E (37)
Tambda
           C (33)
8 (8)
S (14)
                "S (14)
print
"E (35)
"R (38)
"U (41)
"Y (45)
"W (49)
"Z (53)
"G (59)
|id
"_G (65)
"Tambda
"_Z (57)
Tambda
"_Y (48)
"ambda
"_V (48)
Tambda
"_U (44)
Tambda
                         "_R (40)
"lambda
                       "_E (37)
               C (33)
B (8)
S (15)
                 "C (33)
*B (5)
if
```

```
V (48)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         "_U (44)
C (33)
B (8)
S (13)
              (A) S (13) Fotum  
** X (31)  
**E (35)  
**R (38)  
**U (41)  
**U (45)  
**W (49)  
**Z (53)  
**G (59)  
**Id  
**G (59)  
**G (59) 
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Tambda
"_R (40)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               )
S (14)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               print
(
*E (35)
*R (38)
*U (41)
*V (45)
*W (49)
*Z (53)
*G (62)
*CAD
** Z (57)
*Tambda
**_U (48)
**_Tambda
**_U (48)
**_Tambda
**_U (44)
**_Tambda
**_U (44)
**_Tambda
                                                                                                        id
"_G (65)
                                                                                       "_G (65)
Tambda
"_Z (57)
Tambda
"_W (52)
Tambda
"_V (46)
                                                                                            "_V (46)
"W (40)
"Z (53)
"G (61)
"cteent
"_Z (57)
"lambda
"_W (52)
"lambda
"_V (48)
"lambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 Tambda
"_U (44)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                Tambda
"_R (40)
Tambda
"_E (37)
Tambda
                                                                           "_U (44)
lambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 "C (33)
*B (5)
if
                                                                "_R (40)
Tambda
                                                     (* E (3S) *R (38) *U (41) *V (45) *W (49) *2 (53) *G (63) *Ctebool *"_2 (57) *Tambda *"_W (52) *Tambda *"_U (44) *Tambda *"_R (40) *Tambda *"_R (40) *Tambda *"_E (37) *Tambda
                                                                                                          "_W (52)
"lambda
                                                                                                             _V (46)
                                                                                                             W (49)
Z (53)
G (61)
                                                                                                        "_Z (57)
"lambda
"_W (52)
"lambda
"_V (48)
"lambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                S (14)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        print
(*E (35)
*R (38)
*U (41)
*V (45)
*W (49)
*G (52)
*CAD
*Z (57)
*Iambda
*W (52)
*Iambda
***
**Iambda
                                                                                         "_U (44)
                                                                              Tambda
R (40)
                                                                                               lambda
                                                                                    E (36)
                                                                              ** (438)

** (41)

** (41)

** (42)

** (42)

** (40)

** (40)

** (53)

** (650)

** (650)

** (650)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       Tambda
"_V (48)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               "_U (44)
"lambda
                                                                                                                      "_G (65)
lambda
"_Z (57)
lambda
"_W (52)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                "_R (40)
Tambda
"_E (37)
                                                                                                                   lambda
_V (48)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          "C (34)
                                                                                                                lambda
_U (42)
                                                                                                  _U (42)

= V (45)

V (45)

V (46)

Z (53)

G (61)

Cteant

_Z (57)

Tambda

_W (52)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           )
"P (1)
*B (4)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              var
T (11)
                                                                                                                                 "_W (52)
"lambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            ;
P (1)
B (8)
S (12)
                                                                                                                "_V (48)
lambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          id
"_S (19)
                                                                                                        "_U (44)
Tambda
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        (
L (24)
                                                                                                          R (39)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          Tambda
)
                                                                                                  "U (41)
"V (45)
"W (49)
"Z (53)
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 P (3)
```

TABLA PRINCIPAL #1:	
* LEXEMA: 'boolean'	* LEXEMA: 'demo'
ATRIBUTOS:	ATRIBUTOS:
+despl: 0	+numParam: 0
+tipo: 'bool'	+TipoRetorno: 'tipo_vacio'
	+EtiqFuncion: 'ETIdemo5'
* LEXEMA: 'bisiesto'	
ATRIBUTOS:	* LEXEMA: 'A_A_A_'
+numParam: 1	ATRIBUTOS:
+TipoRetorno: 'bool'	+despl: 6
+EtiqFuncion: 'ETIbisiesto2'	+tipo: 'string'
+TipoParam1: 'int'	
+ModoParam1: 1	
	TSL de la funcion 'bisiesto' #2:
* LEXEMA: 'dias'	* LEXEMA: 'a' (parametro de
ATRIBUTOS:	entrada de la funcion)
+numParam: 2	ATRIBUTOS:
+TipoRetorno: 'int'	+despl: 0
+EtiqFuncion: 'ETIdias3'	+tipo: 'int'
+TipoParam1: 'int'	
+ModoParam1: 1	
+TipoParam2: 'int'	TSL de la funcion 'dias' #3:
+ModoParam2: 1	* LEXEMA: 'm' (parametro de
	entrada de la funcion)
* LEXEMA: 'esFechaCorrecta'	ATRIBUTOS:
ATRIBUTOS:	+despl: 0
+numParam: 3	+tipo: 'int'
+TipoRetorno: 'bool'	
+EtiqFuncion:	* LEXEMA: 'a' (parametro de
'ETIesFechaCorrecta4'	entrada de la funcion)
+TipoParam1: 'int'	ATRIBUTOS:
+ModoParam1: 1	+despl: 2
+TipoParam2: 'int'	+tipo: 'int'
+ModoParam2: 1	
+TipoParam3: 'int'	* LEXEMA: 'dd'
+ModoParam3: 1	ATRIBUTOS:

```
+despl: 4
      +tipo: 'int'
      _____
TSL de la funcion 'esFechaCorrecta'
* LEXEMA: 'd' (parametro de
entrada de la funcion)
      ATRIBUTOS:
      +despl: 0
      +tipo: 'int'
* LEXEMA: 'm' (parametro de
entrada de la funcion)
      ATRIBUTOS:
      +despl: 2
      +tipo: 'int'
* LEXEMA: 'a' (parametro de
entrada de la funcion)
      ATRIBUTOS:
      +despl: 4
```

TSL de la funcion 'demo' #5:

\_\_\_\_\_

+tipo: 'int'

## Prueba 5:

```
var bool boolean;
boolean = true;
var string pepe;
function bool bisiesto (int a)
      return (a % 4 > 0 && a - 122 != 0 || a * 400 < 0);
function int dias (int m, int a)
{
      var int dd;
      print ( 'di cuantos dias tiene el mes ' );
      print (m);
      prompt(pepe);
      if (bisiesto(a)) dd = dd / 1;
      return dd;
}
function bool esFechaCorrecta (int d, int m, int a)
{
      return !(d > dias (m, a));
function demo ()
{
      if (esFechaCorrecta(25, 10, 2018)) print ('OK');
var int aaa111;
demo();
```

<pr,9></pr,9>	<pr,8></pr,8>	<div,></div,>	<id,14></id,14>
<pr,2></pr,2>	<pr,1></pr,1>	<cteent,1></cteent,1>	<para,></para,>
<id,10></id,10>	<id,13></id,13>	<fin,></fin,>	<cteent,25></cteent,25>
<fin,></fin,>	<para,></para,>	<pr,7></pr,7>	<sig,></sig,>
<id,10></id,10>	<pr,1></pr,1>	<id,22></id,22>	<cteent,10></cteent,10>
<asig,></asig,>	<id,20></id,20>	<fin,></fin,>	<sig,></sig,>
<ctebool,1></ctebool,1>	<sig,></sig,>	<llavc,></llavc,>	<cteent,2018></cteent,2018>
<fin,></fin,>	<pr,1></pr,1>	<pr,8></pr,8>	<parc,></parc,>
<pr,9></pr,9>	<id,21></id,21>	<pr,2></pr,2>	<parc,></parc,>
<pr,3></pr,3>	<parc,></parc,>	<id,14></id,14>	<pr,12></pr,12>
<id,11></id,11>	<llava,></llava,>	<para,></para,>	<para,></para,>
<fin,></fin,>	<pr,9></pr,9>	<pr,1></pr,1>	<cad,"ok"></cad,"ok">
<pr,8></pr,8>	<pr,1></pr,1>	<id,20></id,20>	<parc,></parc,>
<pr,2></pr,2>	<id,22></id,22>	<sig,></sig,>	<fin,></fin,>
<id,12></id,12>	<fin,></fin,>	<pr,1></pr,1>	<llavc,></llavc,>
<para,></para,>	<pr,12></pr,12>	<id,21></id,21>	<pr,9></pr,9>
<pr,1></pr,1>	<para,></para,>	<sig,></sig,>	<pr,1></pr,1>
<id,20></id,20>	<cad,"di< td=""><td><pr,1></pr,1></td><td><id,16></id,16></td></cad,"di<>	<pr,1></pr,1>	<id,16></id,16>
<parc,></parc,>	cuantos dias	<id,22></id,22>	<fin,></fin,>
<llava,></llava,>	tiene el mes ">	<parc,></parc,>	<id,15></id,15>
<pr,7></pr,7>	<parc,></parc,>	<llava,></llava,>	<para,></para,>
<para,></para,>	<fin,></fin,>	<pr,7></pr,7>	<parc,></parc,>
<id,20></id,20>	<pr,12></pr,12>	<not,></not,>	<fin,></fin,>
<mod,></mod,>	<para,></para,>	<para,></para,>	
<cteent,4></cteent,4>	<id,20></id,20>	<id,20></id,20>	
<opmay,></opmay,>	<parc,></parc,>	<opmay,></opmay,>	
<cteent,0></cteent,0>	<fin,></fin,>	<id,13></id,13>	
<and,></and,>	<pr,13></pr,13>	<para,></para,>	
<id,20></id,20>	<para,></para,>	<id,21></id,21>	
<menos,></menos,>	<id,11></id,11>	<sig,></sig,>	
<cteent,122></cteent,122>	<parc,></parc,>	<id,22></id,22>	
<opdistinto,></opdistinto,>	<fin,></fin,>	<parc,></parc,>	
<cteent,0></cteent,0>	<pr,4></pr,4>	<parc,></parc,>	
<or,></or,>	<para,></para,>	<fin,></fin,>	
<id,20></id,20>	<id,12></id,12>	<llavc,></llavc,>	
<mul,></mul,>	<para,></para,>	<pr,8></pr,8>	
<cteent,400></cteent,400>	<id,21></id,21>	<id,15></id,15>	
<opmen,></opmen,>	<parc,></parc,>	<para,></para,>	
<cteent,0></cteent,0>	<parc,></parc,>	<parc,></parc,>	
<parc,></parc,>	<id,22></id,22>	<llava,></llava,>	
<fin,></fin,>	<asig,></asig,>	<pr,4></pr,4>	
<llavc,></llavc,>	<id,22></id,22>	<para,></para,>	

TABLA PRINCIPAL #1:	+numParam: 0
* LEXEMA: 'boolean'	+TipoRetorno: 'tipo_vacio'
ATRIBUTOS:	+EtiqFuncion: 'ETIdemo5'
+despl: 0	
+tipo: 'bool'	* LEXEMA: 'aaa111'
	ATRIBUTOS:
* LEXEMA: 'pepe'	+despl: 134
ATRIBUTOS:	+tipo: 'int'
+despl: 4	
+tipo: 'string'	
	TSL de la funcion 'bisiesto' #2:
* LEXEMA: 'bisiesto'	* LEXEMA: 'a' (parametro de entrada de
ATRIBUTOS:	la funcion)
+numParam: 1	ATRIBUTOS:
+TipoRetorno: 'bool'	+despl: 0
+EtiqFuncion: 'ETIbisiesto2'	+tipo: 'int'
+TipoParam1: 'int'	· 
+ModoParam1: 1	
	TSL de la funcion 'dias' #3:
* LEXEMA: 'dias'	* LEXEMA: 'm' (parametro de entrada de
ATRIBUTOS:	la funcion)
+numParam: 2	ATRIBUTOS:
+TipoRetorno: 'int'	+despl: 0
+EtiqFuncion: 'ETIdias3'	+tipo: 'int'
+TipoParam1: 'int'	
+ModoParam1: 1	* LEXEMA: 'a' (parametro de entrada de
+TipoParam2: 'int'	la funcion)
+ModoParam2: 1	ATRIBUTOS:
	+despl: 2
* LEXEMA: 'esFechaCorrecta'	+tipo: 'int'
ATRIBUTOS:	
+numParam: 3	* LEXEMA: 'dd'
+TipoRetorno: 'bool'	ATRIBUTOS:
+Hpoketomo. bool +EtiqFuncion:	
'ETlesFechaCorrecta4'	+despl: 4 +tipo: 'int'
+TipoParam1: 'int'	
•	<del></del>
+ModoParam1: 1	
+TipoParam2: 'int' +ModoParam2: 1	
+TipoParam3: 'int'	
+ModoParam3: 1	
* LEXEMA: 'demo'	
ATRIBUTOS:	

TSL de la funcion 'esFechaCorrecta' #4: \* LEXEMA: 'd' (parametro de entrada de la funcion) ATRIBUTOS: +despl: 0 +tipo: 'int' \* LEXEMA: 'm' (parametro de entrada de la funcion) ATRIBUTOS: +despl: 2 +tipo: 'int' \* LEXEMA: 'a' (parametro de entrada de la funcion) ATRIBUTOS: +despl: 4 +tipo: 'int'

TSL de la funcion 'demo' #5:

#### **Pruebas Incorrectas:**

Se muestra la lista de tokens y la tabla de símbolos hasta que se encuentra un error.

## Prueba 6:

```
var bool b;
function bool bisiesto (int a)
{
      return (1 % 4 > 0 && 1 - 1022 != 0 || 1 * 400 < 0);
c=2+2+2+2+2+2+2+2+2+2+2+2+2+2+2+2;
function int dias (int m)
{
      switch (m)
            case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            return 3;
                   break;
            case 4: case 6: while(true){a=2;} case 9: case 11:
                   break; return 2;
            case 2: if (bisiesto (a)) a=1;
            default: a=2;
      }
      return 2;
      b = bisiesto(x);
return b;
```

# Error de compilacion:

Exception('ERROR Semántico en linea 24: sentencia return fuera de funcion')

<pr,9></pr,9>	<id,12></id,12>	<pr,8></pr,8>	<pr,7></pr,7>	<dospuntos,></dospuntos,>
<pr,2></pr,2>	<asig,></asig,>	<pr,1></pr,1>	<cteent,3></cteent,3>	<pr,4></pr,4>
<id,10></id,10>	<cteent,2></cteent,2>	<id,13></id,13>	<fin,></fin,>	<para,></para,>
<fin,></fin,>	<mas,></mas,>	<para,></para,>	<pr,6></pr,6>	<id,11></id,11>
<pr,8></pr,8>	<cteent,2></cteent,2>	<pr,1></pr,1>	<fin,></fin,>	<para,></para,>
<pr,2></pr,2>	<mas,></mas,>	<id,20></id,20>	<pr,11></pr,11>	<id,14></id,14>
<id,11></id,11>	<cteent,2></cteent,2>	<parc,></parc,>	<cteent,4></cteent,4>	<parc,></parc,>
<para,></para,>	<mas,></mas,>	<llava,></llava,>	<dospuntos,></dospuntos,>	<parc,></parc,>
<pr,1></pr,1>	<cteent,2></cteent,2>	<pr,10></pr,10>	<pr,11></pr,11>	<id,14></id,14>
<id,20></id,20>	<mas,></mas,>	<para,></para,>	<cteent,6></cteent,6>	<asig,></asig,>
<parc,></parc,>	<cteent,2></cteent,2>	<id,20></id,20>	<dospuntos,></dospuntos,>	<cteent,1></cteent,1>
<llava,></llava,>	<mas,></mas,>	<parc,></parc,>	<pr,14></pr,14>	<fin,></fin,>
<pr,7></pr,7>	<cteent,2></cteent,2>	<llava,></llava,>	<para,></para,>	<pr,5></pr,5>
<para,></para,>	<mas,></mas,>	<pr,11></pr,11>	<ctebool,1></ctebool,1>	<dospuntos,></dospuntos,>
<cteent,1></cteent,1>	<cteent,2></cteent,2>	<cteent,1></cteent,1>	<parc,></parc,>	<id,14></id,14>
<mod,></mod,>	<mas,></mas,>	<dospuntos,></dospuntos,>	<llava,></llava,>	<asig,></asig,>
<cteent,4></cteent,4>	<cteent,2></cteent,2>	<pr,11></pr,11>	<id,14></id,14>	<cteent,2></cteent,2>
<opmay,></opmay,>	<mas,></mas,>	<cteent,3></cteent,3>	<asig,></asig,>	<fin,></fin,>
<cteent,0></cteent,0>	<cteent,2></cteent,2>	<dospuntos,></dospuntos,>	<cteent,2></cteent,2>	<llavc,></llavc,>
<and,></and,>	<mas,></mas,>	<pr,11></pr,11>	<fin,></fin,>	<pr,7></pr,7>
<cteent,1></cteent,1>	<cteent,2></cteent,2>	<cteent,5></cteent,5>	<llavc,></llavc,>	<cteent,2></cteent,2>
<menos,></menos,>	<mas,></mas,>	<dospuntos,></dospuntos,>	<pr,11></pr,11>	<fin,></fin,>
<cteent,1022></cteent,1022>	<cteent,2></cteent,2>	<pr,11></pr,11>	<cteent,9></cteent,9>	<id,10></id,10>
<opdistinto,></opdistinto,>	<mas,></mas,>	<cteent,7></cteent,7>	<dospuntos,></dospuntos,>	<asig,></asig,>
<cteent,0></cteent,0>	<cteent,2></cteent,2>	<dospuntos,></dospuntos,>	<pr,11></pr,11>	<id,11></id,11>
<or,></or,>	<mas,></mas,>	<pr,11></pr,11>	<cteent,11></cteent,11>	<para,></para,>
<cteent,1></cteent,1>	<cteent,2></cteent,2>	<cteent,8></cteent,8>	<dospuntos,></dospuntos,>	<id,15></id,15>
<mul,></mul,>	<mas,></mas,>	<dospuntos,></dospuntos,>	<pr,6></pr,6>	<parc,></parc,>
<cteent,400></cteent,400>	<cteent,2></cteent,2>	<pr,11></pr,11>	<fin,></fin,>	<fin,></fin,>
<opmen,></opmen,>	<mas,></mas,>	<cteent,10></cteent,10>	<pr,7></pr,7>	<llavc,></llavc,>
<cteent,0></cteent,0>	<cteent,2></cteent,2>	<dospuntos,></dospuntos,>	<cteent,2></cteent,2>	<pr,7></pr,7>
<parc,></parc,>	<mas,></mas,>	<pr,11></pr,11>	<fin,></fin,>	<id,10></id,10>
<fin,></fin,>	<cteent,2></cteent,2>	<cteent,12></cteent,12>	<pr,11></pr,11>	
<llavc,></llavc,>	<fin,></fin,>	<dospuntos,></dospuntos,>	<cteent,2></cteent,2>	

TABLA PRINCIPAL #1:  * LEXEMA: 'b'  ATRIBUTOS:  +despl: 0  +tipo: 'bool'	* LEXEMA: 'a'  ATRIBUTOS:  +despl: 6  +tipo: 'int'
	* LEXEMA: 'x'
* LEXEMA: 'bisiesto'	ATRIBUTOS:
ATRIBUTOS:	+despl: 8
+numParam: 1	+tipo: 'int'
+TipoRetorno: 'bool'	
+EtiqFuncion: 'ETIbisiesto2'	
+TipoParam1: 'int'	TSL de la funcion 'bisiesto' #2:
+ModoParam1: 1	* LEXEMA: 'a' (parametro de entrada de
	la funcion)
* LEXEMA: 'c'	ATRIBUTOS:
ATRIBUTOS:	+despl: 0
+despl: 4	+tipo: 'int'
+tipo: 'int'	
* LEXEMA: 'dias'	TSL de la funcion 'dias' #3:
ATRIBUTOS:	* LEXEMA: 'm' (parametro de entrada de
+numParam: 1	la funcion)
+TipoRetorno: 'int'	ATRIBUTOS:
+EtiqFuncion: 'ETIdias3'	+despl: 0
+TipoParam1: 'int'	+tipo: 'int'
+ModoParam1: 1	

## Prueba 7:

# Error de compilacion:

Exception('ERROR Lexico en linea 16: Entero supera el tamaño máximo')

<pr,9></pr,9>	<menos,></menos,>	<mas,></mas,>
<pr,2></pr,2>	<cteent,1022></cteent,1022>	<cteent,2></cteent,2>
<id,10></id,10>	<opdistinto,></opdistinto,>	<mas,></mas,>
<fin,></fin,>	<cteent,0></cteent,0>	<cteent,2></cteent,2>
<id,11></id,11>	<or,></or,>	<mas,></mas,>
<asig,></asig,>	<cteent,1></cteent,1>	<cteent,2></cteent,2>
<cteent,2></cteent,2>	<mul,></mul,>	<mas,></mas,>
<fin,></fin,>	<cteent,400></cteent,400>	<cteent,2></cteent,2>
<pr,8></pr,8>	<opmen,></opmen,>	<mas,></mas,>
<pr,2></pr,2>	<cteent,0></cteent,0>	<cteent,2></cteent,2>
<id,12></id,12>	<parc,></parc,>	<mas,></mas,>
<para,></para,>	<fin,></fin,>	<cteent,2></cteent,2>
<pr,2></pr,2>	<llavc,></llavc,>	<mas,></mas,>
<id,20></id,20>	<id,11></id,11>	<cteent,2></cteent,2>
<parc,></parc,>	<asig,></asig,>	<mas,></mas,>
<llava,></llava,>	<cteent,1></cteent,1>	<cteent,2></cteent,2>
<pr,9></pr,9>	<fin,></fin,>	<mas,></mas,>
<pr,2></pr,2>	<id,10></id,10>	<cteent,2></cteent,2>
<id,21></id,21>	<asig,></asig,>	<mas,></mas,>
<fin,></fin,>	<id,12></id,12>	<cteent,2></cteent,2>
<id,21></id,21>	<para,></para,>	<mas,></mas,>
<asig,></asig,>	<id,12></id,12>	<cteent,2></cteent,2>
<ctebool,1></ctebool,1>	<para,></para,>	<mas,></mas,>
<fin,></fin,>	<id,12></id,12>	<cteent,2></cteent,2>
<pr,7></pr,7>	<para,></para,>	<mas,></mas,>
<para,></para,>	<ctebool,1></ctebool,1>	<cteent,2></cteent,2>
<cteent,1></cteent,1>	<parc,></parc,>	<mas,></mas,>
<mod,></mod,>	<parc,></parc,>	<cteent,2></cteent,2>
<cteent,4></cteent,4>	<parc,></parc,>	<mas,></mas,>
<opmay,></opmay,>	<fin,></fin,>	<cteent,2></cteent,2>
<cteent,0></cteent,0>	<id,11></id,11>	<fin,></fin,>
<and,></and,>	<asig,></asig,>	<id,11></id,11>
<cteent,1></cteent,1>	<cteent,2></cteent,2>	<asig,></asig,>

```
TABLA PRINCIPAL #1:
* LEXEMA: 'b'
      ATRIBUTOS:
      +despl: 0
      +tipo: 'bool'
* LEXEMA: 'c'
      ATRIBUTOS:
      +despl: 4
      +tipo: 'int'
     _____
* LEXEMA: 'bisiesto'
      ATRIBUTOS:
      +numParam: 1
      +TipoRetorno: 'bool'
      +EtiqFuncion: 'ETIbisiesto2'
      +TipoParam1: 'bool'
      +ModoParam1: 1
     _____
TSL de la funcion 'bisiesto' #2:
* LEXEMA: 'a' (parametro de entrada de la funcion)
      ATRIBUTOS:
      +despl: 0
      +tipo: 'bool'
* LEXEMA: 'c'
      ATRIBUTOS:
      +despl: 4
      +tipo: 'bool'
```

# Prueba 8:

```
var int a; var bool b;
b = false && 2 == 2%2 || true && a > 34 + 3*a;
var bool c;
c = a > a;
if (c) a -= 1;
a=2+(2+c);
if (2 < a) a -= 4;
a = a + 4;
print (a);
print (b);</pre>
```

## Error de compilacion:

Exception('ERROR semantico en linea 6: expresion incorrecta')

## Tokens

<pr,9></pr,9>
<pr,1></pr,1>
<id,10></id,10>
<fin,></fin,>
<pr,9></pr,9>
<pr,2></pr,2>
<id,11></id,11>
<fin,></fin,>
<id,11></id,11>
<asig,></asig,>
<ctebool,0></ctebool,0>
<and,></and,>
<cteent,2></cteent,2>
<opig,></opig,>
<cteent,2></cteent,2>
<mod,></mod,>
<cteent,2></cteent,2>
<or,></or,>
<ctebool,1></ctebool,1>
<and,></and,>
<id,10></id,10>
<opmay,></opmay,>
<cteent,34></cteent,34>
<mas,></mas,>
<cteent,3></cteent,3>

<MUL,>

<ID,10> <FIN,> <PR,9> <PR,2> <ID,12> <FIN,> <ID,12> <ASIG,> <ID,10> <OPMAY,> <ID,10> <FIN,> <PR,4> <PARA,> <ID,12> <PARC,> <ID,10> <ASIGR,> <cteent,1> <FIN,> <ID,10> <ASIG,> <cteent,2> <MAS,> <PARA,> <cteent,2> <MAS,> <ID,12> <PARC,>

# TABLA PRINCIPAL #1: \* LEXEMA: 'a' ATRIBUTOS: +despl: 0 +tipo: 'int' ----- \* LEXEMA: 'b' ATRIBUTOS: +despl: 2 +tipo: 'bool' ----- \* LEXEMA: 'c' ATRIBUTOS: +despl: 6 +tipo: 'bool'

## Prueba 9:

```
var bool boolean;
function bool bisiesto (int a)
{
      return (a + 4 > 0 && a + 100 != 0 && a % 400 != 0);
function int dias (int m, int a)
      var int dd;
      print ( 'di cuantos dias tiene el mes ' );
      print (m);
      prompt(dd);
      if (bisiesto(a,boolean)) dd = dd % 1;
      return dd;
}
function bool esFechaCorrecta (int d, int m, int a)
      return m > 1 || m > 12 || d == 1 && d < dias (m, a);
function demo ()
{
      if (esFechaCorrecta(25, 10, 2018)) print ('OK');
      if (true) print ('OK');
}
var string A_A_A_;
demo();
```

# Error de compilacion:

Exception("ERROR Semántico en linea 12: llamada a función que pide ['int'] con parámetro incorrectos['int', 'bool']")

<pr,9></pr,9>	<and,></and,>	<para,></para,>
<pr,2></pr,2>	<id,20></id,20>	<cad,"di cuantos<="" td=""></cad,"di>
<id,10></id,10>	<mod,></mod,>	dias tiene el mes ">
<fin,></fin,>	<cteent,400></cteent,400>	<parc,></parc,>
<pr,8></pr,8>	<opdistinto,></opdistinto,>	<fin,></fin,>
<pr,2></pr,2>	<cteent,0></cteent,0>	<pr,12></pr,12>
<id,11></id,11>	<parc,></parc,>	<para,></para,>
<para,></para,>	<fin,></fin,>	<id,20></id,20>
<pr,1></pr,1>	<llavc,></llavc,>	<parc,></parc,>
<id,20></id,20>	<pr,8></pr,8>	<fin,></fin,>
<parc,></parc,>	<pr,1></pr,1>	<pr,13></pr,13>
<llava,></llava,>	<id,12></id,12>	<para,></para,>
<pr,7></pr,7>	<para,></para,>	<id,22></id,22>
<para,></para,>	<pr,1></pr,1>	<parc,></parc,>
<id,20></id,20>	<id,20></id,20>	<fin,></fin,>
<mas,></mas,>	<sig,></sig,>	<pr,4></pr,4>
<cteent,4></cteent,4>	<pr,1></pr,1>	<para,></para,>
<opmay,></opmay,>	<id,21></id,21>	<id,11></id,11>
<cteent,0></cteent,0>	<parc,></parc,>	<para,></para,>
<and,></and,>	<llava,></llava,>	<id,21></id,21>
<id,20></id,20>	<pr,9></pr,9>	<sig,></sig,>
<mas,></mas,>	<pr,1></pr,1>	<id,10></id,10>
<cteent,100></cteent,100>	<id,22></id,22>	<parc,></parc,>
<opdistinto,></opdistinto,>	<fin,></fin,>	<parc,></parc,>
<cteent,0></cteent,0>	<pr,12></pr,12>	

TABLA PRINCIPAL #1:  * LEXEMA: 'boolean'  ATRIBUTOS:  +despl: 0  +tipo: 'bool'	TSL de la funcion 'bisiesto' #2:  * LEXEMA: 'a' (parametro de entrada de la funcion)  ATRIBUTOS:  +despl: 0
* LEXEMA: 'bisiesto' ATRIBUTOS:	+tipo: 'int'
+numParam: 1	
+TipoRetorno: 'bool'	TSL de la funcion 'dias' #3:
+EtiqFuncion: 'ETIbisiesto2'	* LEXEMA: 'm' (parametro de
+TipoParam1: 'int'	entrada de la funcion)
+ModoParam1: 1	ATRIBUTOS:
	+despl: 0
* LEXEMA: 'dias'	+tipo: 'int'
ATRIBUTOS:	
+numParam: 2	* LEXEMA: 'a' (parametro de
+TipoRetorno: 'int'	entrada de la funcion)
+EtiqFuncion: 'ETIdias3'	ATRIBUTOS:
+TipoParam1: 'int'	+despl: 2
+ModoParam1: 1	+tipo: 'int'
+TipoParam2: 'int'	
+ModoParam2: 1	* LEXEMA: 'dd'
	ATRIBUTOS:
	+despl: 4
	+tipo: 'int'

#### Prueba 10:

```
var int contador;
var bool esCierto;
a=2;
var string cadena;

function int divide (int num1 ,string num2, int num3)
{
    a = b; //variable no existente que se declarará como global y entera
    var int b; //declaracion de variable local
    b=a; //b coge el valor de la variable global
    var int a; //declaracion de variable local de mismo nombre de la global
    que hace que esta ultima no sea ya accesible

return a;
ab=a3;
a3+1;
divide(1,'a',3);
}
```

# Error de compilacion:

Exception('ERROR Sintactico en linea 15: sintaxis incorrecta')

<pr,9></pr,9>	<id,22></id,22>
<pr,1></pr,1>	<parc,></parc,>
<id,10></id,10>	<llava,></llava,>
<fin,></fin,>	<id,12></id,12>
<pr,9></pr,9>	<asig,></asig,>
<pr,2></pr,2>	<id,15></id,15>
<id,11></id,11>	<fin,></fin,>
<fin,></fin,>	<pr,9></pr,9>
<id,12></id,12>	<pr,1></pr,1>
<asig,></asig,>	<id,23></id,23>
<cteent,2></cteent,2>	<fin,></fin,>
<fin,></fin,>	<id,23></id,23>
<pr,9></pr,9>	<asig,></asig,>
<pr,3></pr,3>	<id,12></id,12>
<id,13></id,13>	<fin,></fin,>
<fin,></fin,>	<pr,9></pr,9>
<pr,8></pr,8>	<pr,1></pr,1>
<pr,1></pr,1>	<id,24></id,24>
<id,14></id,14>	<fin,></fin,>
<para,></para,>	<pr,7></pr,7>
<pr,1></pr,1>	<id,24></id,24>
<id,20></id,20>	<fin,></fin,>
<sig,></sig,>	<id,16></id,16>
<pr,3></pr,3>	<asig,></asig,>
<id,21></id,21>	<id,17></id,17>
<sig,></sig,>	<fin,></fin,>
<pr,1></pr,1>	<id,17></id,17>
	<mas,></mas,>

TABLA PRINCIPAL #1:  * LEXEMA: 'contador'  ATRIBUTOS:  +despl: 0  +tipo: 'int'	* LEXEMA: 'ab' ATRIBUTOS: +despl: 138 +tipo: 'int'
* LEXEMA: 'esCierto'  ATRIBUTOS:  +despl: 2  +tipo: 'bool'   * LEXEMA: 'a'  ATRIBUTOS:	* LEXEMA: 'a3' ATRIBUTOS: +despl: 140 +tipo: 'int'
+despl: 6	TSL de la funcion 'divide' #2:  * LEXEMA: 'num1' (parametro de entrada de la funcion)  ATRIBUTOS:  +despl: 0  +tipo: 'int'
* LEXEMA: 'divide' ATRIBUTOS: +numParam: 3 +TipoRetorno: 'int' +EtiqFuncion: 'ETIdivide2' +TipoParam1: 'int' +ModoParam1: 1 +TipoParam2: 'string' +ModoParam3: 1 +TipoParam3: 'int' +ModoParam3: 1	* LEXEMA: 'num2' (parametro de entrada de la funcion)  ATRIBUTOS:  +despl: 2  +tipo: 'string'   * LEXEMA: 'num3' (parametro de entrada de la funcion)  ATRIBUTOS:  +despl: 130  +tipo: 'int'
* LEXEMA: 'b' ATRIBUTOS: +despl: 136 +tipo: 'int'	* LEXEMA: 'b'  ATRIBUTOS:  +despl: 132  +tipo: 'int'   * LEXEMA: 'a'  ATRIBUTOS:  +despl: 134  +tipo: 'int'