**PROCESADORES DE LENGUAJES**



**Grupo 64**

Christian Paniagua Paniagua x150380

Javier Pérez Martín x150147

Pablo Heras Aranzana x150046

**Índice**

[**1.** **Diseño Analizador Léxico** 4](#_Toc535856417)

[1.1.Tokens 4](#_Toc535856418)

[1.2.Gramática 5](#_Toc535856419)

[1.3.Autómata 6](#_Toc535856420)

[**2.** **Diseño Analizador Sintáctico** 8](#_Toc535856421)

[2.1. Gramática 8](#_Toc535856422)

[2.2. Demostración LL(1) 9](#_Toc535856423)

[2.3.Procedimientos 10](#_Toc535856424)

[**3.** **Diseño Analizador Semántico** 14](#_Toc535856425)

[**Anexo** 15](#_Toc535856426)

# **Diseño Analizador Léxico**

## 1.1.Tokens

**<cteent, número>**

**<CAD, “lexema”>**

**<ctebool, 1>** (true)

**<ctebool, 0>** (false)

**<MAS, ->** (+)

**<MENOS, ->** (-)

**<MUL, ->** (\*)

**<DIV, ->** (/)

**<MOD, ->** (%)

**<AND, ->** (&&)

**<LLAVA, ->** ( { )

**<LLAVC, ->** ( } )

**<PARA, ->** ( ( )

**<PARC, ->** ( ) )

**<FIN, - >** ( ; )

**<SIG, - >** ( , )

**<OR, ->** (||)

**<NOT, ->** ( ! )

**<ASIGR, ->** (-=)

**<ASIG, - >** (=)

**<OPMAY, - >** (>)

**<OPMEN, - >** (<)

**<OPIG, - >** (==)

**<OPDISTINTO,- >** (!=)

**<DOSPUNTOS, - >** (:)

**<PR, 1 >** (int)

**<PR, 2 >** (bool)

**<PR, 3 >** (string)

**<PR, 4 >** (if)

**<PR, 5 >** (default)

**<PR, 6 >** (break)

**<PR, 7 >** (return)

**<PR, 8 >** (function)

**<PR, 9 >** (var)

**<PR, 10 >** (switch)

**<PR, 11 >** (case)

**<PR, 12 >** (print)

**<PR, 13 >** (prompt)

**<Id, pos\_ts>** (siendo “pos\_ts” la posición que ocupa el id dentro de la tabla de símbolos)

# 1.2.Gramática

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

C --> \ C' | oc1 C | '

C' --> 't' C | 'n' C | ' C | " C | \ C

D --> = | λ

D1 --> = | λ

D2 --> = | λ

D3 --> = | λ

D4 --> = | λ

E --> d E | λ

I --> l I | d I | \_ I | λ

J --> &

K --> '|'

N --> \* N' | λ

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

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

Definiciones:

del={esp, tab, eol}

d = {0..9}

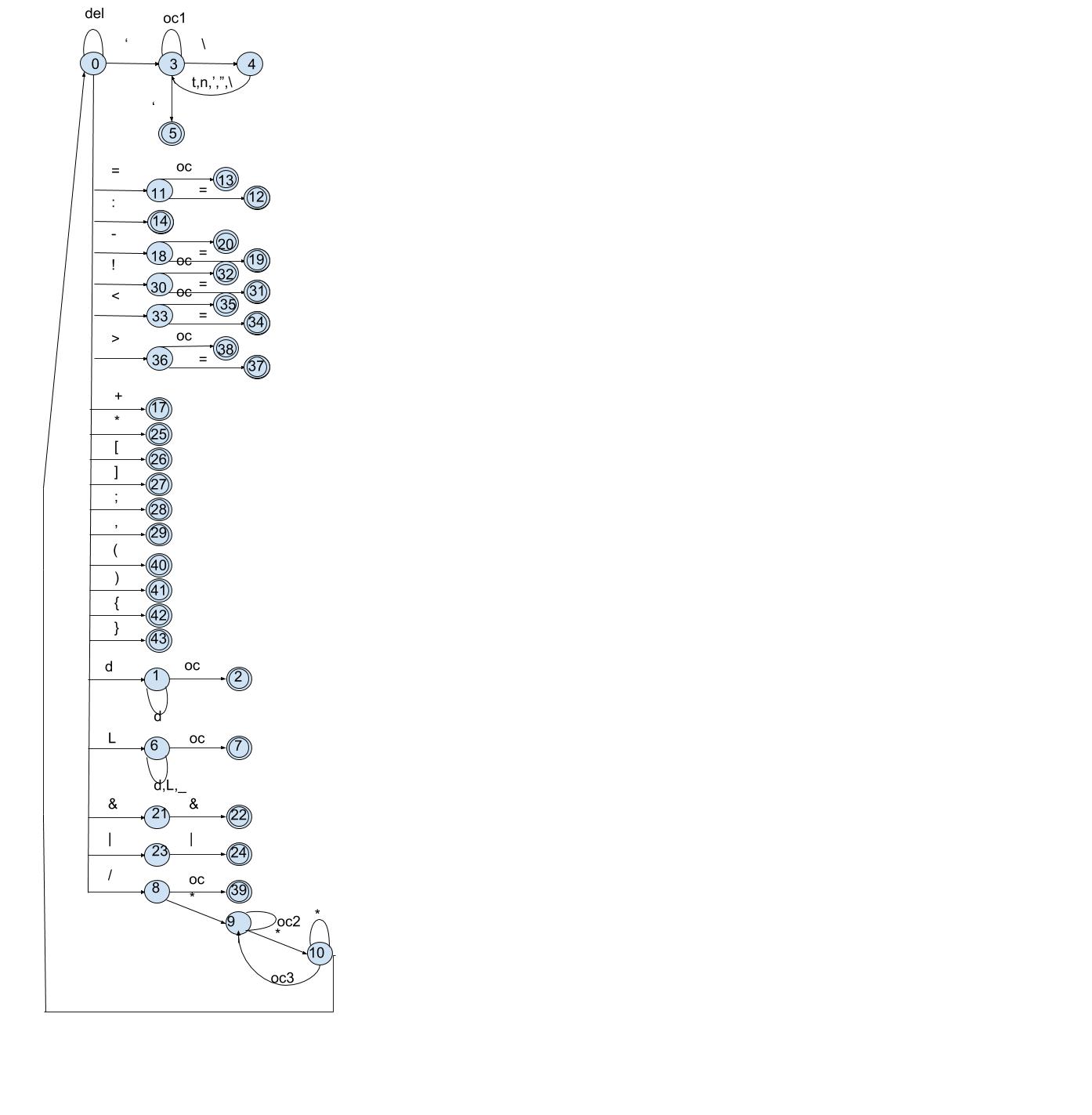
l = {a..z, A..Z}

oc1 = {cualquier carácter imprimible} - {\}

oc2 = {cualquier carácter imprimible} - {\*}

oc3 = {cualquier carácter imprimible} - {/,\*}

## 1.3.Autómata



1.4.Acciones semánticas

L=Leer\_carácter()

0:1 -->L, valor = valor\_ascii(d)

1:1 -->L, valor = valor\*10 + valor\_ascii(d)

1:2 -->if( valor > 32767) then **ERROR**

else **G.Token(<ENT, valor>)**

0:3 -->L,lexema = carácter\_ascii( ‘ )

3:3 -->L,lexema = lexema ⊕ carácter ascii(oc1)

3:4 -->L,lexema = lexema ⊕ carácter ascii( \ )

4:3 --> L,lexema = lexema ⊕ {

carácter ascii( t )

carácter ascii( n )

carácter ascii( ‘ )

carácter ascii( “ )

carácter ascii( \ )

}

3:5-->L, lexema = lexema ⊕ carácter ascii( ‘ )

0:6-->L, lexema = carácter\_asciii( l )

6:6-->L, lexema = lexema ⊕ {

carácter ascii( l )

carácter ascii( d )

carácter ascii( -)

}

6:7--> If (pos = TS(lexema)

then G.token(< Id , pos>)

Else if ( pos = T.palreservada(lexema))

then **G.token(<PR , pos>)**

else pos = Insertar Ts(lexema),

Gtoken(< Id, pos>)

0:8-->L

0:39-->**G.token(<DIV, ~ >)**

8:9-->L

9:9-->L

9:10-->L

10:10-->L

10:0-->L

0:11-->L

11:12-->L, **G.token(<OP\_IG, pos>)**

11:13-->**G.token(<Asig, ~ >)**

0:14-->L, **G.token(<Dos\_puntos, ~ >)**

0:17-->L, **G.token(<Mas, ~ >)**

0:18-->L

18:19-->L, **G.token(<Asig\_R, ~ >)**

18:20--> **G.token(<Menos, ~ >)**

0:21-->L

21:22-->L, **G.token(<AND, ~ >)**

0:23-->L

23:24-->L, **G.token(<OR, ~ >)**

0:25-->L, **G.token(<MUL, ~ >)**

0:26-->L, **G.token(<CORA, ~ >)**

0:27-->L, **G.token(<CORC, ~ >)**

0:28-->L, **G.token(<FIN, ~ >)**

0:29-->L,

0:30-->L

30:31-->L, **G.token(<OP\_DISTINTO, ~ >)**

30:32-->L, **G.token(<NOT, ~ >)**

0:33-->L

33:34-->L, **G.token(<OP\_MEN\_IG, ~ >)**

33:35-->L, **G.token(<OP\_MEN, ~ >)**

0:36-->L

36:38-->L, **G.token(<OP\_MAY\_IG, ~ >)**

36:37-->L, **G.token(<OP\_MAY, ~ >)**

0:40-->L, **G.token(<PARA, ~ >)**

0:41-->L, **G.token(<PARC, ~ >)**

0:42-->L, **G.token(<LLAVA, ~ >)**

0:43-->L, **G.token(<LLAVC, ~ >)**

# **Diseño Analizador Sintáctico**

## Gramática

Terminales = { var id ; if ( ) while } { switch case break default int bool string return print

prompt function , || && == != < > -= ! = + - \* / % CAD : cteent ctebool }

NoTerminales = { P B T S \_S F H L Q A K X C E \_E R \_R U \_U V \_V W \_W Z \_Z G \_G D }

Axioma = P

Producciones = {

P -> B P

P -> F P

P -> lambda

B -> var T id ;

B -> if ( E ) S

B -> while ( E ) { C }

B -> switch ( E ) { D }

B -> S

T -> int

T -> bool

T -> string

S -> id \_S

S -> return X ;

S -> print ( E ) ;

S -> prompt ( id ) ;

S -> break ;

\_S -> = E ;

\_S -> -= E ;

\_S -> ( L ) ;

F -> function H id ( A ) { C }

H -> T

H -> lambda

L -> E Q

L -> lambda

Q -> , E Q

Q -> lambda

A -> T id K

A -> lambda

K -> , T id K

K -> lambda

X -> E

X -> lambda

C -> B C

C -> lambda

E -> R \_E

\_E -> || R \_E

\_E -> lambda

R -> U \_R

\_R -> && U \_R

\_R -> lambda

U -> V \_U

\_U -> == V \_U

\_U -> != V \_U

\_U -> lambda

V -> W \_V

\_V -> > W \_V

\_V -> < W \_V

\_V -> lambda

W -> Z \_W

\_W -> + Z \_W

\_W -> - Z \_W

\_W -> lambda

Z -> G \_Z

\_Z -> \* G \_Z

\_Z -> / G \_Z

\_Z -> % G \_Z

\_Z -> lambda

G -> ! G

G -> id \_G

G -> ( E )

G -> cteent

G -> CAD

G -> ctebool

\_G -> ( L )

\_G -> lambda

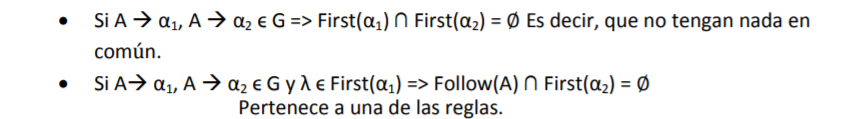
D -> case cteent : C D

D -> default : C

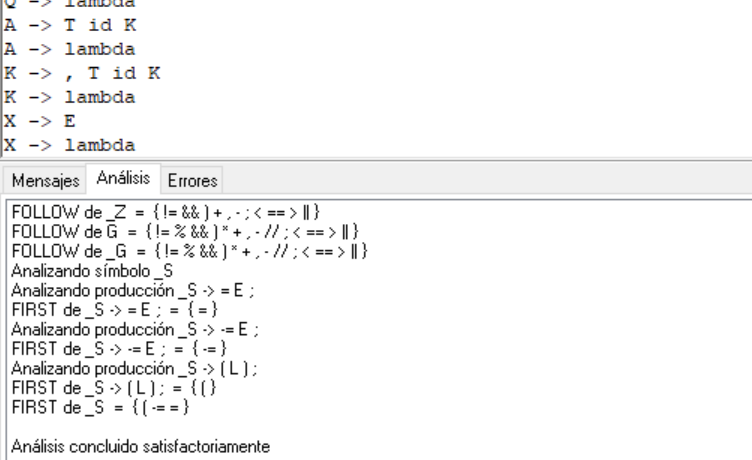
}

## 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):



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).



## 2.3.Procedimientos

A continuación, mostramos el código de los procedimientos que realizarán el análisis sintáctico. El resultado de éste se volcará en el fichero de parse.

print("Des ");

Proc P;

if(st ϵ First(B P)={ break id if print prompt return switch var while }) then

{

print("1");

B();

P();

}

else if(st ϵ First(F P)={ function }) then

{

print("2");

F();

P();

}

else if(st ϵ Follow(P)={ $ })

{

print("3");

}

else ERROR;

END;

Proc B;

if(st ϵ First(var T id ;)={ var }) then

{

print("4");

EqT(var);

T();

EqT(id);

EqT(;);

}

else if(st ϵ First(if ( E ) S)={ if }) then

{

print("5");

EqT(if);

EqT(();

E();

EqT());

S();

}

else if(st ϵ First(while ( E ) { C })={ while }) then

{

print("6");

EqT(while);

EqT(();

E();

EqT());

EqT({);

C();

EqT(});

}

else if(st ϵ First(switch ( E ) { D })={ switch }) then

{

print("7");

EqT(switch);

EqT(();

E();

EqT());

EqT({);

D();

EqT(});

}

else if(st ϵ First(S)={ break id print prompt return }) then

{

print("8");

S();

}

else ERROR;

END;

Proc T;

if(st ϵ First(int)={ int }) then

{

print("9");

EqT(int);

}

else if(st ϵ First(bool)={ bool }) then

{

print("10");

EqT(bool);

}

else if(st ϵ First(string)={ string }) then

{

print("11");

EqT(string);

}

else ERROR;

END;

Proc S;

if(st ϵ First(id S')={ id }) then

{

print("12");

EqT(id);

S'();

}

else if(st ϵ First(return X ;)={ return }) then

{

print("13");

EqT(return);

X();

EqT(;);

}

else if(st ϵ First(print ( E ) ;)={ print }) then

{

print("14");

EqT(print);

EqT(();

E();

EqT());

EqT(;);

}

else if(st ϵ First(prompt ( id ) ;)={ prompt }) then

{

print("15");

EqT(prompt);

EqT(();

EqT(id);

EqT());

EqT(;);

}

else if(st ϵ First( break ;)={ break }) then

{

print("16");

EqT(break);

EqT(;);

}

else ERROR;

END;

Proc S';

if(st ϵ First(= E ;)={ = }) then

{

print("17");

EqT(=);

E();

EqT(;);

}

else if(st ϵ First(-= E ;)={ -= }) then

{

print("18");

EqT(-=);

E();

EqT(;);

}

else if(st ϵ First(( L ) ;)={ ( }) then

{

print("19");

EqT(();

L();

EqT());

EqT(;);

}

else ERROR;

END;

Proc F;

if(st ϵ First(function H id ( A ) { C })={ function }) then

{

print("20");

EqT(function);

H();

EqT(id);

EqT(();

A();

EqT());

EqT({);

C();

EqT(});

}

else ERROR;

END;

Proc H;

if(st ϵ First(T)={ int bool string }) then

{

print("21");

T();

}

else if(st ϵ Follow(H)={ id }) then

{

print("22");

}

else ERROR;

END;

Proc L;

if(st ϵ First(E Q)={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("23");

E();

Q();

}

else if(st ϵ Follow(L)={ ) }) then

{

print("24");

}

else ERROR;

END;

Proc Q;

if(st ϵ First(, E Q)={ , }) then

{

print("25");

EqT(,);

E();

Q();

}

else if(st ϵ Follow(Q)={ ) }) then

{

print("26");

}

else ERROR;

END;

Proc A;

if(st ϵ First(T id K)={ int bool string }) then

{

print("27");

T();

EqT(id);

K();

}

else if(st ϵ Follow(A)={ ) ; }) then

{

print("28");

}

else ERROR;

END;

Proc K;

if(st ϵ First(, T id K)={ , }) then

{

print("29");

EqT(,);

T();

EqT(id);

K();

}

else if(st ϵ Follow(K)={ ) }) then

{

print("30");

}

else ERROR;

END;

Proc X;

if(st ϵ First(E)={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("31");

E();

}

else if(st ϵ Follow(X)={ ; }) then

{

print("32");

}

else ERROR;

END;

Proc C;

if(st ϵ First(B C)={ break id if print prompt return switch var while }) then

{

print("33");

B();

C();

}

else if(st ϵ Follow(C)={ case } default }) then

{

print("34");

}

else ERROR;

END;

Proc E;

if(st ϵ First(R E')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("35");

R();

E'();

}

else ERROR;

END;

Proc E';

if(st ϵ First(|| R E')={ || }) then

{

print("36");

EqT(||);

R();

E'();

}

else if(st ϵ Follow(E')={ ) , ; }) then

{

print("37");

}

else ERROR;

END;

Proc R;

if(st ϵ First(U R')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("38");

U();

R'();

}

else ERROR;

END;

Proc R';

if(st ϵ First(&& U R')={ && }) then

{

print("39");

EqT(&&);

U();

R'();

}

else if(st ϵ Follow(R')={ ) , ; || }) then

{

print("40");

}

else ERROR;

END;

Proc U;

if(st ϵ First(V U')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("41");

V();

U'();

}

else ERROR;

END;

Proc U';

if(st ϵ First(== V U')={ == }) then

{

print("42");

EqT(==);

V();

U'();

}

else if(st ϵ First(!= V U')={ != }) then

{

print("43");

EqT(!=);

V();

U'();

}

else if(st ϵ Follow(U')={ && ) , ; || }) then

{

print("44");

}

else ERROR;

END;

Proc V;

if(st ϵ First(W V')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("45");

W();

V'();

}

else ERROR;

END;

Proc V';

if(st ϵ First(> W V')={ > }) then

{

print("46");

EqT(>);

W();

V'();

}

else if(st ϵ First(< W V')={ < }) then

{

print("47");

EqT(<);

W();

V'();

}

else if(st ϵ Follow(V')={ != && ) , ; == || }) then

{

print("48");

}

else ERROR;

END;

Proc W;

if(st ϵ First(Z W')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("49");

Z();

W'();

}

else ERROR;

END;

Proc W';

if(st ϵ First(+ Z W')={ + }) then

{

print("50");

EqT(+);

Z();

W'();

}

else if(st ϵ First(- Z W')={ - }) then

{

print("51");

EqT(-);

Z();

W'();

}

else if(st ϵ Follow(W')={ != && ) , ; < == > || }) then

{

print("52");

}

else ERROR;

END;

Proc Z;

if(st ϵ First(G Z')={ ! ( cte\_bool CAD cte\_ent id }) then

{

print("53");

G();

Z'();

}

else ERROR;

END;

Proc Z';

if(st ϵ First(\* G Z')={ \* }) then

{

print("54");

EqT(\*);

G();

Z'();

}

else if(st ϵ First(/ G Z')={ / }) then

{

print("55");

EqT(/);

G();

Z'();

}

else if(st ϵ First(% G Z')={ % }) then

{

print("56");

EqT(%);

G();

Z'();

}

else if(st ϵ Follow(Z')={ != && ) + , - ; < == > || }) then

{

print("57");

}

else ERROR;

END;

Proc G;

if(st ϵ First(! G)={ ! }) then

{

print("58");

EqT(!);

G();

}

else if(st ϵ First(id G')={ id }) then

{

print("59");

EqT(id);

G'();

}

else if(st ϵ First(( E ))={ ( }) then

{

print("60");

EqT(();

E();

EqT());

}

else if(st ϵ First(cte\_ent)={ cte\_ent }) then

{

print("61");

EqT(cte\_ent);

}

else if(st ϵ First(CAD)={ CAD }) then

{

print("62");

EqT(CAD);

}

else if(st ϵ First(cte\_bool)={ cte\_bool }) then

{

print("63");

EqT(cte\_bool);

}

else ERROR;

END;

Proc G';

if(st ϵ First(( L ))={ ( }) then

{

print("64");

EqT(();

L();

EqT());

}

else if(st ϵ Follow(G')={ != % && ) \* + , - / ; < == > || }) then

{

print("65");

}

else ERROR;

END;

Proc D;

if(st ϵ First(case cte\_ent : C D)={ case }) then

{

print("66");

EqT(case);

EqT(cte\_ent);

EqT(:);

C();

D();

}

else if(st ϵ ) then

{

print("67");

EqT(default);

EqT(:);

C();

}

else ERROR;

END;

# **Diseño Analizador Semántico**

# **Anexo**

**Pruebas Correctas:**

**Prueba 1:**

var string texto;/\* comentariooooooooooooooooooooooooooo

ooooooooooooooooooooooooooo

\*/

//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,3>

<ID,10>

<FIN,>

<PR,8>

<PR,3>

<ID,11>

<PARA,>

<PARC,>

<LLAVA,>

<PR,12>

<PARA,>

<CAD,"Introduce un texto">

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,10>

<PARC,>

<FIN,>

<PR,7>

<CAD,"cad">

<FIN,>

<LLAVC,>

<ID,10>

<ASIG,>

<ID,11>

<PARA,>

<PARC,>

<FIN,>

<PR,8>

<ID,12>

<PARA,>

<PR,3>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<SIG,>

<PR,1>

<ID,22>

<PARC,>

<LLAVA,>

<PR,12>

<PARA,>

<ID,20>

<PARC,>

<FIN,>

<LLAVC,>

<ID,11>

<PARA,>

<PARC,>

<FIN,>

<PR,9>

<PR,3>

<ID,13>

<FIN,>

<ID,13>

<ASIG,>

<ID,13>

<FIN,>

Árbol

Des 1 4 11 2 20 21 11 28 33 8 14 35 38 41 45 49 53 62 57 52 48 44 40 37 33 8 15 33 8 13 31 35 38 41 45 49 53 62 57 52 48 44 40 37 34 1 8 12 17 35 38 41 45 49 53 59 64 24 57 52 48 44 40 37 2 20 22 27 11 29 9 29 9 30 33 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 34 1 8 12 19 24 1 4 11 1 8 12 17 35 38 41 45 49 53 59 65 57 52 48 44 40 37 3

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'texto'

ATRIBUTOS:

+despl: 0

+tipo: 'string'

-------------------------------

\* LEXEMA: 'pideTexto'

ATRIBUTOS:

+numParam: 0

+TipoRetorno: 'string'

+EtiqFuncion: 'ETIpideTexto2'

-------------------------------

\* LEXEMA: 'imprime'

ATRIBUTOS:

+numParam: 3

+TipoRetorno: 'tipo\_vacio'

+EtiqFuncion: 'ETIimprime3'

+TipoParam1: 'string'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

+TipoParam3: 'int'

+ModoParam3: 1

-------------------------------

\* LEXEMA: 'textoAux'

ATRIBUTOS:

+despl: 130

+tipo: 'string'

-------------------------------

TSL de la funcion 'pideTexto' #2:

TSL de la funcion 'imprime' #3:

\* LEXEMA: 'msg' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'string'

-------------------------------

\* LEXEMA: 'p' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 128

+tipo: 'int'

-------------------------------

\* LEXEMA: 'p1' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 130

+tipo: 'int'

-----------------------------

**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);

Tokens

<PR,9>

<PR,1>

<ID,10>

<FIN,>

<PR,9>

<PR,2>

<ID,11>

<FIN,>

<ID,11>

<ASIG,>

<ctebool,0>

<AND,>

<cteent,2>

<OPIG,>

<cteent,2>

<MOD,>

<cteent,2>

<OR,>

<ctebool,1>

<AND,>

<ID,10>

<OPMAY,>

<cteent,34>

<MAS,>

<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,>

<PR,4>

<PARA,>

<cteent,2>

<OPMEN,>

<ID,10>

<PARC,>

<ID,10>

<ASIGR,>

<cteent,4>

<FIN,>

<ID,10>

<ASIG,>

<ID,10>

<MAS,>

<cteent,4>

<FIN,>

<PR,12>

<PARA,>

<ID,10>

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<ID,11>

<PARC,>

<FIN,>

Árbol

Des 1 4 9 1 4 10 1 8 12 17 35 38 41 45 49 53 63 57 52 48 44 39 41 45 49 53 61 57 52 48 42 45 49 53 61 56 61 57 52 48 44 40 36 38 41 45 49 53 63 57 52 48 44 39 41 45 49 53 59 65 57 52 46 49 53 61 57 50 53 61 54 59 65 57 52 48 44 40 37 1 4 10 1 8 12 17 35 38 41 45 49 53 59 65 57 52 46 49 53 59 65 57 52 48 44 40 37 1 5 35 38 41 45 49 53 59 65 57 52 48 44 40 37 12 18 35 38 41 45 49 53 61 57 52 48 44 40 37 1 5 35 38 41 45 49 53 61 57 52 47 49 53 59 65 57 52 48 44 40 37 12 18 35 38 41 45 49 53 61 57 52 48 44 40 37 1 8 12 17 35 38 41 45 49 53 59 65 57 50 53 61 57 52 48 44 40 37 1 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 1 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 3

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,1>

<ID,10>

<FIN,>

<PR,9>

<PR,1>

<ID,11>

<FIN,>

<PR,9>

<PR,1>

<ID,12>

<FIN,>

<PR,12>

<PARA,>

<CAD,"Introduce el primer operando">

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,10>

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<CAD,"Introduce el segundo operando">

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,11>

<PARC,>

<FIN,>

<PR,8>

<PR,1>

<ID,13>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<PARC,>

<LLAVA,>

<PR,7>

<ID,20>

<DIV,>

<ID,21>

<FIN,>

<LLAVC,>

<ID,12>

<ASIG,>

<ID,13>

<PARA,>

<ID,10>

<SIG,>

<ID,11>

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<ID,12>

<PARC,>

<FIN,>

Árbol

Des 1 4 9 1 4 9 1 4 9 1 8 14 35 38 41 45 49 53 62 57 52 48 44 40 37 1 8 15 1 8 14 35 38 41 45 49 53 62 57 52 48 44 40 37 1 8 15 2 20 21 9 27 9 29 9 30 33 8 13 31 35 38 41 45 49 53 59 65 55 59 65 57 52 48 44 40 37 34 1 8 12 17 35 38 41 45 49 53 59 64 23 35 38 41 45 49 53 59 65 57 52 48 44 40 37 25 35 38 41 45 49 53 59 65 57 52 48 44 40 37 26 57 52 48 44 40 37 1 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 3

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,2>

<ID,10>

<FIN,>

<PR,8>

<PR,2>

<ID,11>

<PARA,>

<PR,1>

<ID,20>

<PARC,>

<LLAVA,>

<PR,7>

<PARA,>

<ID,20>

<MAS,>

<cteent,4>

<OPMAY,>

<cteent,0>

<AND,>

<ID,20>

<MAS,>

<cteent,100>

<OPDISTINTO,>

<cteent,0>

<AND,>

<ID,20>

<MOD,>

<cteent,400>

<OPDISTINTO,>

<cteent,0>

<PARC,>

<FIN,>

<LLAVC,>

<PR,8>

<PR,1>

<ID,12>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<PARC,>

<LLAVA,>

<PR,9>

<PR,1>

<ID,22>

<FIN,>

<PR,12>

<PARA,>

<CAD,"di cuantos dias tiene el mes ">

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<ID,20>

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,22>

<PARC,>

<FIN,>

<PR,4>

<PARA,>

<ID,11>

<PARA,>

<ID,21>

<PARC,>

<PARC,>

<ID,22>

<ASIG,>

<ID,22>

<MOD,>

<cteent,1>

<FIN,>

<PR,7>

<ID,22>

<FIN,>

<LLAVC,>

<PR,8>

<PR,2>

<ID,13>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<SIG,>

<PR,1>

<ID,22>

<PARC,>

<LLAVA,>

<PR,7>

<ID,21>

<OPMAY,>

<cteent,1>

<OR,>

<ID,21>

<OPMAY,>

<cteent,12>

<OR,>

<ID,20>

<OPIG,>

<cteent,1>

<AND,>

<ID,20>

<OPMEN,>

<ID,12>

<PARA,>

<ID,21>

<SIG,>

<ID,22>

<PARC,>

<FIN,>

<LLAVC,>

<PR,8>

<ID,14>

<PARA,>

<PARC,>

<LLAVA,>

<PR,4>

<PARA,>

<ID,13>

<PARA,>

<cteent,25>

<SIG,>

<cteent,10>

<SIG,>

<cteent,2018>

<PARC,>

<PARC,>

<PR,12>

<PARA,>

<CAD,"OK">

<PARC,>

<FIN,>

<PR,4>

<PARA,>

<ctebool,1>

<PARC,>

<PR,12>

<PARA,>

<CAD,"OK">

<PARC,>

<FIN,>

<LLAVC,>

<PR,9>

<PR,3>

<ID,15>

<FIN,>

<ID,14>

<PARA,>

<PARC,>

<FIN,>

Árbol

Des 1 4 10 2 20 21 10 27 9 30 33 8 13 31 35 38 41 45 49 53 60 35 38 41 45 49 53 59 65 57 50 53 61 57 52 46 49 53 61 57 52 48 44 39 41 45 49 53 59 65 57 50 53 61 57 52 48 43 45 49 53 61 57 52 48 44 39 41 45 49 53 59 65 56 61 57 52 48 43 45 49 53 61 57 52 48 44 40 37 57 52 48 44 40 37 34 2 20 21 9 27 9 29 9 30 33 4 9 33 8 14 35 38 41 45 49 53 62 57 52 48 44 40 37 33 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 33 8 15 33 5 35 38 41 45 49 53 59 64 23 35 38 41 45 49 53 59 65 57 52 48 44 40 37 26 57 52 48 44 40 37 12 17 35 38 41 45 49 53 59 65 56 61 57 52 48 44 40 37 33 8 13 31 35 38 41 45 49 53 59 65 57 52 48 44 40 37 34 2 20 21 10 27 9 29 9 29 9 30 33 8 13 31 35 38 41 45 49 53 59 65 57 52 46 49 53 61 57 52 48 44 40 36 38 41 45 49 53 59 65 57 52 46 49 53 61 57 52 48 44 40 36 38 41 45 49 53 59 65 57 52 48 42 45 49 53 61 57 52 48 44 39 41 45 49 53 59 65 57 52 47 49 53 59 64 23 35 38 41 45 49 53 59 65 57 52 48 44 40 37 25 35 38 41 45 49 53 59 65 57 52 48 44 40 37 26 57 52 48 44 40 37 34 2 20 22 28 33 5 35 38 41 45 49 53 59 64 23 35 38 41 45 49 53 61 57 52 48 44 40 37 25 35 38 41 45 49 53 61 57 52 48 44 40 37 25 35 38 41 45 49 53 61 57 52 48 44 40 37 26 57 52 48 44 40 37 14 35 38 41 45 49 53 62 57 52 48 44 40 37 33 5 35 38 41 45 49 53 63 57 52 48 44 40 37 14 35 38 41 45 49 53 62 57 52 48 44 40 37 34 1 4 11 1 8 12 19 24 3

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'boolean'

ATRIBUTOS:

+despl: 0

+tipo: 'bool'

-------------------------------

\* LEXEMA: 'bisiesto'

ATRIBUTOS:

+numParam: 1

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIbisiesto2'

+TipoParam1: 'int'

+ModoParam1: 1

-------------------------------

\* LEXEMA: 'dias'

ATRIBUTOS:

+numParam: 2

+TipoRetorno: 'int'

+EtiqFuncion: 'ETIdias3'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

-------------------------------

\* LEXEMA: 'esFechaCorrecta'

ATRIBUTOS:

+numParam: 3

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIesFechaCorrecta4'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

+TipoParam3: 'int'

+ModoParam3: 1

-------------------------------

\* LEXEMA: 'demo'

ATRIBUTOS:

+numParam: 0

+TipoRetorno: 'tipo\_vacio'

+EtiqFuncion: 'ETIdemo5'

-------------------------------

\* LEXEMA: 'A\_A\_A\_'

ATRIBUTOS:

+despl: 6

+tipo: 'string'

-------------------------------

TSL de la funcion 'bisiesto' #2:

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

TSL de la funcion 'dias' #3:

\* LEXEMA: 'm' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 2

+tipo: 'int'

-------------------------------

\* LEXEMA: 'dd'

ATRIBUTOS:

+despl: 4

+tipo: 'int'

-------------------------------

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:

**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();

Tokens

<PR,9>

<PR,2>

<ID,10>

<FIN,>

<ID,10>

<ASIG,>

<ctebool,1>

<FIN,>

<PR,9>

<PR,3>

<ID,11>

<FIN,>

<PR,8>

<PR,2>

<ID,12>

<PARA,>

<PR,1>

<ID,20>

<PARC,>

<LLAVA,>

<PR,7>

<PARA,>

<ID,20>

<MOD,>

<cteent,4>

<OPMAY,>

<cteent,0>

<AND,>

<ID,20>

<MENOS,>

<cteent,122>

<OPDISTINTO,>

<cteent,0>

<OR,>

<ID,20>

<MUL,>

<cteent,400>

<OPMEN,>

<cteent,0>

<PARC,>

<FIN,>

<LLAVC,>

<PR,8>

<PR,1>

<ID,13>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<PARC,>

<LLAVA,>

<PR,9>

<PR,1>

<ID,22>

<FIN,>

<PR,12>

<PARA,>

<CAD,"di cuantos dias tiene el mes ">

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<ID,20>

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,11>

<PARC,>

<FIN,>

<PR,4>

<PARA,>

<ID,12>

<PARA,>

<ID,21>

<PARC,>

<PARC,>

<ID,22>

<ASIG,>

<ID,22>

<DIV,>

<cteent,1>

<FIN,>

<PR,7>

<ID,22>

<FIN,>

<LLAVC,>

<PR,8>

<PR,2>

<ID,14>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<SIG,>

<PR,1>

<ID,22>

<PARC,>

<LLAVA,>

<PR,7>

<NOT,>

<PARA,>

<ID,20>

<OPMAY,>

<ID,13>

<PARA,>

<ID,21>

<SIG,>

<ID,22>

<PARC,>

<PARC,>

<FIN,>

<LLAVC,>

<PR,8>

<ID,15>

<PARA,>

<PARC,>

<LLAVA,>

<PR,4>

<PARA,>

<ID,14>

<PARA,>

<cteent,25>

<SIG,>

<cteent,10>

<SIG,>

<cteent,2018>

<PARC,>

<PARC,>

<PR,12>

<PARA,>

<CAD,"OK">

<PARC,>

<FIN,>

<LLAVC,>

<PR,9>

<PR,1>

<ID,16>

<FIN,>

<ID,15>

<PARA,>

<PARC,>

<FIN,>

Árbol

Des 1 4 10 1 8 12 17 35 38 41 45 49 53 63 57 52 48 44 40 37 1 4 11 2 20 21 10 27 9 30 33 8 13 31 35 38 41 45 49 53 60 35 38 41 45 49 53 59 65 56 61 57 52 46 49 53 61 57 52 48 44 39 41 45 49 53 59 65 57 51 53 61 57 52 48 43 45 49 53 61 57 52 48 44 40 36 38 41 45 49 53 59 65 54 61 57 52 47 49 53 61 57 52 48 44 40 37 57 52 48 44 40 37 34 2 20 21 9 27 9 29 9 30 33 4 9 33 8 14 35 38 41 45 49 53 62 57 52 48 44 40 37 33 8 14 35 38 41 45 49 53 59 65 57 52 48 44 40 37 33 8 15 33 5 35 38 41 45 49 53 59 64 23 35 38 41 45 49 53 59 65 57 52 48 44 40 37 26 57 52 48 44 40 37 12 17 35 38 41 45 49 53 59 65 55 61 57 52 48 44 40 37 33 8 13 31 35 38 41 45 49 53 59 65 57 52 48 44 40 37 34 2 20 21 10 27 9 29 9 29 9 30 33 8 13 31 35 38 41 45 49 53 58 60 35 38 41 45 49 53 59 65 57 52 46 49 53 59 64 23 35 38 41 45 49 53 59 65 57 52 48 44 40 37 25 35 38 41 45 49 53 59 65 57 52 48 44 40 37 26 57 52 48 44 40 37 57 52 48 44 40 37 34 2 20 22 28 33 5 35 38 41 45 49 53 59 64 23 35 38 41 45 49 53 61 57 52 48 44 40 37 25 35 38 41 45 49 53 61 57 52 48 44 40 37 25 35 38 41 45 49 53 61 57 52 48 44 40 37 26 57 52 48 44 40 37 14 35 38 41 45 49 53 62 57 52 48 44 40 37 34 1 4 9 1 8 12 19 24 3

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'boolean'

ATRIBUTOS:

+despl: 0

+tipo: 'bool'

-------------------------------

\* LEXEMA: 'pepe'

ATRIBUTOS:

+despl: 4

+tipo: 'string'

-------------------------------

\* LEXEMA: 'bisiesto'

ATRIBUTOS:

+numParam: 1

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIbisiesto2'

+TipoParam1: 'int'

+ModoParam1: 1

-------------------------------

\* LEXEMA: 'dias'

ATRIBUTOS:

+numParam: 2

+TipoRetorno: 'int'

+EtiqFuncion: 'ETIdias3'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

-------------------------------

\* LEXEMA: 'esFechaCorrecta'

ATRIBUTOS:

+numParam: 3

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIesFechaCorrecta4'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

+TipoParam3: 'int'

+ModoParam3: 1

-------------------------------

\* LEXEMA: 'demo'

ATRIBUTOS:

+numParam: 0

+TipoRetorno: 'tipo\_vacio'

+EtiqFuncion: 'ETIdemo5'

-------------------------------

\* LEXEMA: 'aaa111'

ATRIBUTOS:

+despl: 134

+tipo: 'int'

-------------------------------

TSL de la funcion 'bisiesto' #2:

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

TSL de la funcion 'dias' #3:

\* LEXEMA: 'm' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 2

+tipo: 'int'

-------------------------------

\* LEXEMA: 'dd'

ATRIBUTOS:

+despl: 4

+tipo: 'int'

-------------------------------

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;

Mensaje error

Error de compilacion:

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

Tokens

<PR,9>

<PR,2>

<ID,10>

<FIN,>

<PR,8>

<PR,2>

<ID,11>

<PARA,>

<PR,1>

<ID,20>

<PARC,>

<LLAVA,>

<PR,7>

<PARA,>

<cteent,1>

<MOD,>

<cteent,4>

<OPMAY,>

<cteent,0>

<AND,>

<cteent,1>

<MENOS,>

<cteent,1022>

<OPDISTINTO,>

<cteent,0>

<OR,>

<cteent,1>

<MUL,>

<cteent,400>

<OPMEN,>

<cteent,0>

<PARC,>

<FIN,>

<LLAVC,>

<ID,12>

<ASIG,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<FIN,>

<PR,8>

<PR,1>

<ID,13>

<PARA,>

<PR,1>

<ID,20>

<PARC,>

<LLAVA,>

<PR,10>

<PARA,>

<ID,20>

<PARC,>

<LLAVA,>

<PR,11>

<cteent,1>

<DOSPUNTOS,>

<PR,11>

<cteent,3>

<DOSPUNTOS,>

<PR,11>

<cteent,5>

<DOSPUNTOS,>

<PR,11>

<cteent,7>

<DOSPUNTOS,>

<PR,11>

<cteent,8>

<DOSPUNTOS,>

<PR,11>

<cteent,10>

<DOSPUNTOS,>

<PR,11>

<cteent,12>

<DOSPUNTOS,>

<PR,7>

<cteent,3>

<FIN,>

<PR,6>

<FIN,>

<PR,11>

<cteent,4>

<DOSPUNTOS,>

<PR,11>

<cteent,6>

<DOSPUNTOS,>

<PR,14>

<PARA,>

<ctebool,1>

<PARC,>

<LLAVA,>

<ID,14>

<ASIG,>

<cteent,2>

<FIN,>

<LLAVC,>

<PR,11>

<cteent,9>

<DOSPUNTOS,>

<PR,11>

<cteent,11>

<DOSPUNTOS,>

<PR,6>

<FIN,>

<PR,7>

<cteent,2>

<FIN,>

<PR,11>

<cteent,2>

<DOSPUNTOS,>

<PR,4>

<PARA,>

<ID,11>

<PARA,>

<ID,14>

<PARC,>

<PARC,>

<ID,14>

<ASIG,>

<cteent,1>

<FIN,>

<PR,5>

<DOSPUNTOS,>

<ID,14>

<ASIG,>

<cteent,2>

<FIN,>

<LLAVC,>

<PR,7>

<cteent,2>

<FIN,>

<ID,10>

<ASIG,>

<ID,11>

<PARA,>

<ID,15>

<PARC,>

<FIN,>

<LLAVC,>

<PR,7>

<ID,10>

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'b'

ATRIBUTOS:

+despl: 0

+tipo: 'bool'

-------------------------------

\* LEXEMA: 'bisiesto'

ATRIBUTOS:

+numParam: 1

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIbisiesto2'

+TipoParam1: 'int'

+ModoParam1: 1

-------------------------------

\* LEXEMA: 'c'

ATRIBUTOS:

+despl: 4

+tipo: 'int'

-------------------------------

\* LEXEMA: 'dias'

ATRIBUTOS:

+numParam: 1

+TipoRetorno: 'int'

+EtiqFuncion: 'ETIdias3'

+TipoParam1: 'int'

+ModoParam1: 1

-------------------------------

\* LEXEMA: 'a'

ATRIBUTOS:

+despl: 6

+tipo: 'int'

-------------------------------

\* LEXEMA: 'x'

ATRIBUTOS:

+despl: 8

+tipo: 'int'

-------------------------------

TSL de la funcion 'bisiesto' #2:

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

TSL de la funcion 'dias' #3:

\* LEXEMA: 'm' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

**Prueba 7:**

var bool b;

c=2;

function bool bisiesto (bool a)

{

var bool c;

c=true;

return (1 % 4 > 0 && 1 - 1022 != 0 || 1 \* 400 < 0);

}

c=1;

b=bisiesto(bisiesto(bisiesto(true)));

c=2+2+2+2+2+2+2+2+2+2+2+2+2+2+2+2;

c=9999999999999999999999999;

Mensaje error

Error de compilacion:

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

Tokens

<PR,9>

<PR,2>

<ID,10>

<FIN,>

<ID,11>

<ASIG,>

<cteent,2>

<FIN,>

<PR,8>

<PR,2>

<ID,12>

<PARA,>

<PR,2>

<ID,20>

<PARC,>

<LLAVA,>

<PR,9>

<PR,2>

<ID,21>

<FIN,>

<ID,21>

<ASIG,>

<ctebool,1>

<FIN,>

<PR,7>

<PARA,>

<cteent,1>

<MOD,>

<cteent,4>

<OPMAY,>

<cteent,0>

<AND,>

<cteent,1>

<MENOS,>

<cteent,1022>

<OPDISTINTO,>

<cteent,0>

<OR,>

<cteent,1>

<MUL,>

<cteent,400>

<OPMEN,>

<cteent,0>

<PARC,>

<FIN,>

<LLAVC,>

<ID,11>

<ASIG,>

<cteent,1>

<FIN,>

<ID,10>

<ASIG,>

<ID,12>

<PARA,>

<ID,12>

<PARA,>

<ID,12>

<PARA,>

<ctebool,1>

<PARC,>

<PARC,>

<PARC,>

<FIN,>

<ID,11>

<ASIG,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<MAS,>

<cteent,2>

<FIN,>

<ID,11>

<ASIG,>

Tabla de Símbolos

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);

Mensaje error

Error de compilacion:

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

Tokens

<PR,9>

<PR,1>

<ID,10>

<FIN,>

<PR,9>

<PR,2>

<ID,11>

<FIN,>

<ID,11>

<ASIG,>

<ctebool,0>

<AND,>

<cteent,2>

<OPIG,>

<cteent,2>

<MOD,>

<cteent,2>

<OR,>

<ctebool,1>

<AND,>

<ID,10>

<OPMAY,>

<cteent,34>

<MAS,>

<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 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 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();

Mensaje error

Error de compilacion:

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

Tokens

<PR,9>

<PR,2>

<ID,10>

<FIN,>

<PR,8>

<PR,2>

<ID,11>

<PARA,>

<PR,1>

<ID,20>

<PARC,>

<LLAVA,>

<PR,7>

<PARA,>

<ID,20>

<MAS,>

<cteent,4>

<OPMAY,>

<cteent,0>

<AND,>

<ID,20>

<MAS,>

<cteent,100>

<OPDISTINTO,>

<cteent,0>

<AND,>

<ID,20>

<MOD,>

<cteent,400>

<OPDISTINTO,>

<cteent,0>

<PARC,>

<FIN,>

<LLAVC,>

<PR,8>

<PR,1>

<ID,12>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,1>

<ID,21>

<PARC,>

<LLAVA,>

<PR,9>

<PR,1>

<ID,22>

<FIN,>

<PR,12>

<PARA,>

<CAD,"di cuantos dias tiene el mes ">

<PARC,>

<FIN,>

<PR,12>

<PARA,>

<ID,20>

<PARC,>

<FIN,>

<PR,13>

<PARA,>

<ID,22>

<PARC,>

<FIN,>

<PR,4>

<PARA,>

<ID,11>

<PARA,>

<ID,21>

<SIG,>

<ID,10>

<PARC,>

<PARC,>

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'boolean'

ATRIBUTOS:

+despl: 0

+tipo: 'bool'

-------------------------------

\* LEXEMA: 'bisiesto'

ATRIBUTOS:

+numParam: 1

+TipoRetorno: 'bool'

+EtiqFuncion: 'ETIbisiesto2'

+TipoParam1: 'int'

+ModoParam1: 1

-------------------------------

\* LEXEMA: 'dias'

ATRIBUTOS:

+numParam: 2

+TipoRetorno: 'int'

+EtiqFuncion: 'ETIdias3'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'int'

+ModoParam2: 1

-------------------------------

TSL de la funcion 'bisiesto' #2:

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

TSL de la funcion 'dias' #3:

\* LEXEMA: 'm' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

\* LEXEMA: 'a' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 2

+tipo: 'int'

-------------------------------

\* 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);

}

Mensaje error

Error de compilacion:

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

Tokens

<PR,9>

<PR,1>

<ID,10>

<FIN,>

<PR,9>

<PR,2>

<ID,11>

<FIN,>

<ID,12>

<ASIG,>

<cteent,2>

<FIN,>

<PR,9>

<PR,3>

<ID,13>

<FIN,>

<PR,8>

<PR,1>

<ID,14>

<PARA,>

<PR,1>

<ID,20>

<SIG,>

<PR,3>

<ID,21>

<SIG,>

<PR,1>

<ID,22>

<PARC,>

<LLAVA,>

<ID,12>

<ASIG,>

<ID,15>

<FIN,>

<PR,9>

<PR,1>

<ID,23>

<FIN,>

<ID,23>

<ASIG,>

<ID,12>

<FIN,>

<PR,9>

<PR,1>

<ID,24>

<FIN,>

<PR,7>

<ID,24>

<FIN,>

<ID,16>

<ASIG,>

<ID,17>

<FIN,>

<ID,17>

<MAS,>

Tabla de Símbolos

TABLA PRINCIPAL #1:

\* LEXEMA: 'contador'

ATRIBUTOS:

+despl: 0

+tipo: 'int'

-------------------------------

\* LEXEMA: 'esCierto'

ATRIBUTOS:

+despl: 2

+tipo: 'bool'

-------------------------------

\* LEXEMA: 'a'

ATRIBUTOS:

+despl: 6

+tipo: 'int'

-------------------------------

\* LEXEMA: 'cadena'

ATRIBUTOS:

+despl: 8

+tipo: 'string'

-------------------------------

\* LEXEMA: 'divide'

ATRIBUTOS:

+numParam: 3

+TipoRetorno: 'int'

+EtiqFuncion: 'ETIdivide2'

+TipoParam1: 'int'

+ModoParam1: 1

+TipoParam2: 'string'

+ModoParam2: 1

+TipoParam3: 'int'

+ModoParam3: 1

-------------------------------

\* LEXEMA: 'b'

ATRIBUTOS:

+despl: 136

+tipo: 'int'

-------------------------------

\* LEXEMA: 'ab'

ATRIBUTOS:

+despl: 138

+tipo: 'int'

-------------------------------

\* LEXEMA: 'a3'

ATRIBUTOS:

+despl: 140

+tipo: 'int'

-------------------------------

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: 'string'

-------------------------------

\* LEXEMA: 'num3' (parametro de entrada de la funcion)

ATRIBUTOS:

+despl: 130

+tipo: 'int'

-------------------------------

\* LEXEMA: 'b'

ATRIBUTOS:

+despl: 132

+tipo: 'int'

-------------------------------

\* LEXEMA: 'a'

ATRIBUTOS:

+despl: 134

+tipo: 'int'

-------------------------------