PROCESADORES DE LENGUAJES



# Christian Paniagua Paniagua x150380

# Javier Pérez Martín x150147

# Pablo Heras Aranzana x150046

## **Tokens**

<Ent, valor>

<CAD, lexema>

<Bool, valor > (true)

<Mas, -> (+)

<Menos, -> (-)

<MUL, -> (\*)

<DIV, -> (/)

<AND, -> (&&)

<LLAVA, -> ( { )

<LLAVC, -> ( } )

<PARA, -> ( ( )

<PARC, -> ( ) )

<CORA, -> ( [ )

<CORC, -> ( ] )

<FIN, - > ( ; )

<SIG, - > ( , )

<OR, -> (||)

<NOT, -> ( ! )

<Asig\_R, -> (-=)

<Asig, - > (=)

<OP\_sal, - > (print (c))

<OP-en, - > (promt (c))

<OP\_MAY, - > (>)

<OP\_MEN, - > (<)

<OP\_MEN\_IG, - > (<=)

<OP\_MAY\_IG, - > (>=)

<OP\_IG, - > (==)

<Dos\_puntos, - > (:)

<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 > (promt)

<Id, ptr\_ts>

# **Gramática**

S --> del S | d E | 'C | l I | / N | = D | : D1 | + | - D2 | & J | '|' K | \* | [ | ] | ; | , | ! D3 | < D4 | D5 >

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

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

D --> = | λ

D1 --> = | λ

D2 --> = | λ

D3 --> = | λ

D4 --> = | λ

D5 --> = | 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}

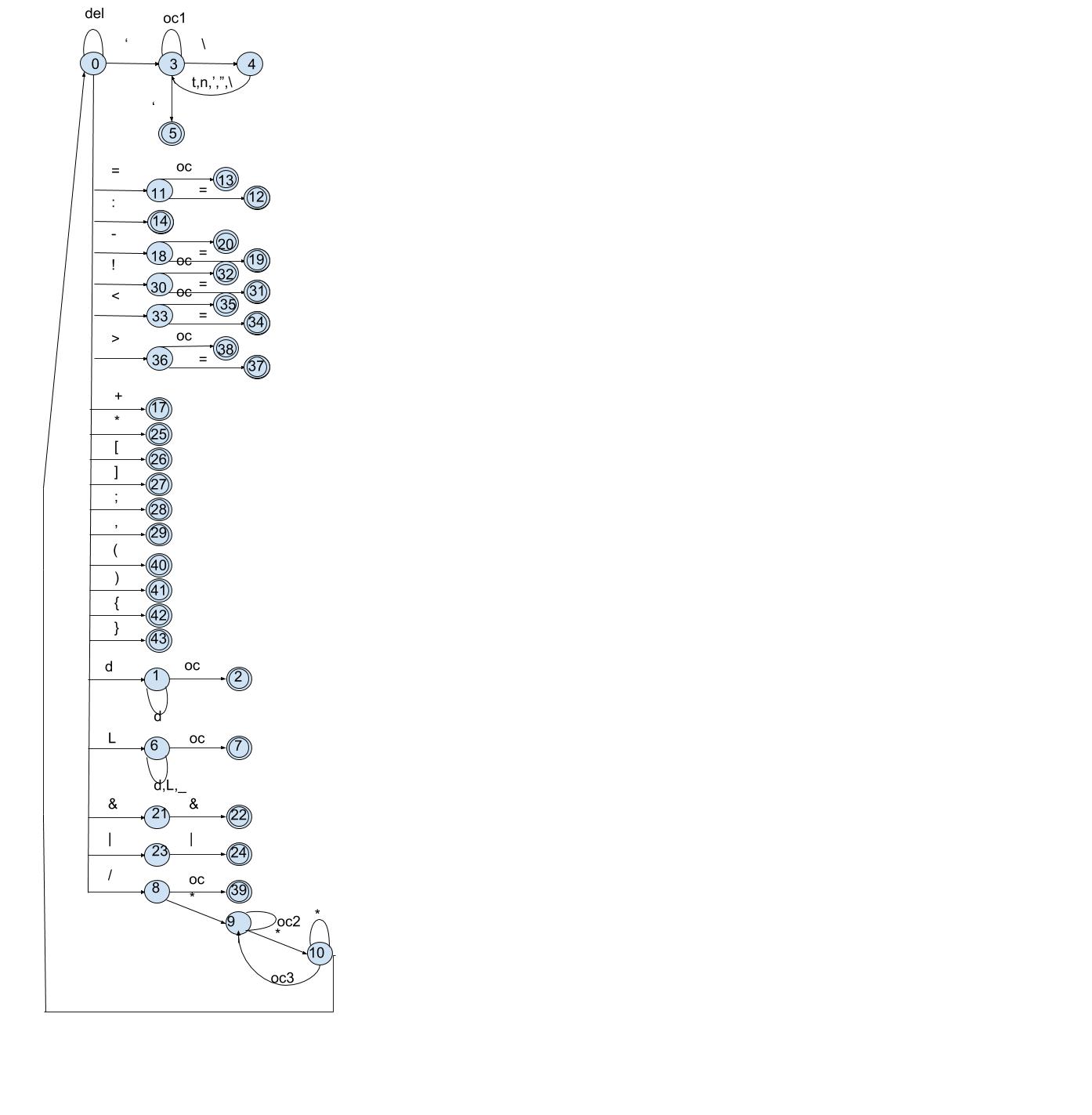
l = {a..z, A..Z}

oc1 = {cualquier carácter imprimible} - {\}

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

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

## **Autómata**



## **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 > 65535) 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(<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, ~ >)**