

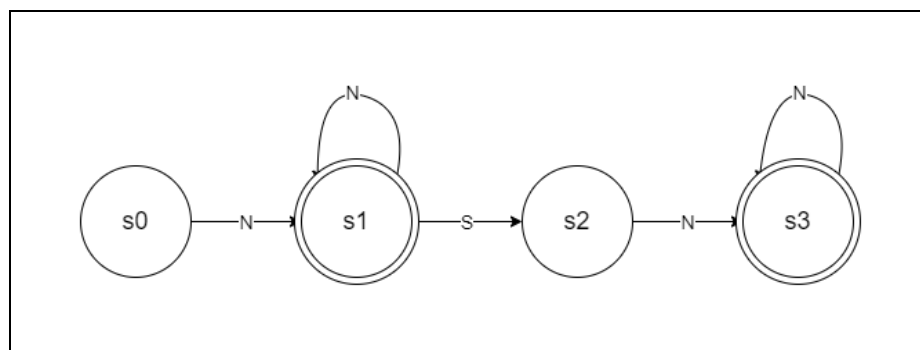
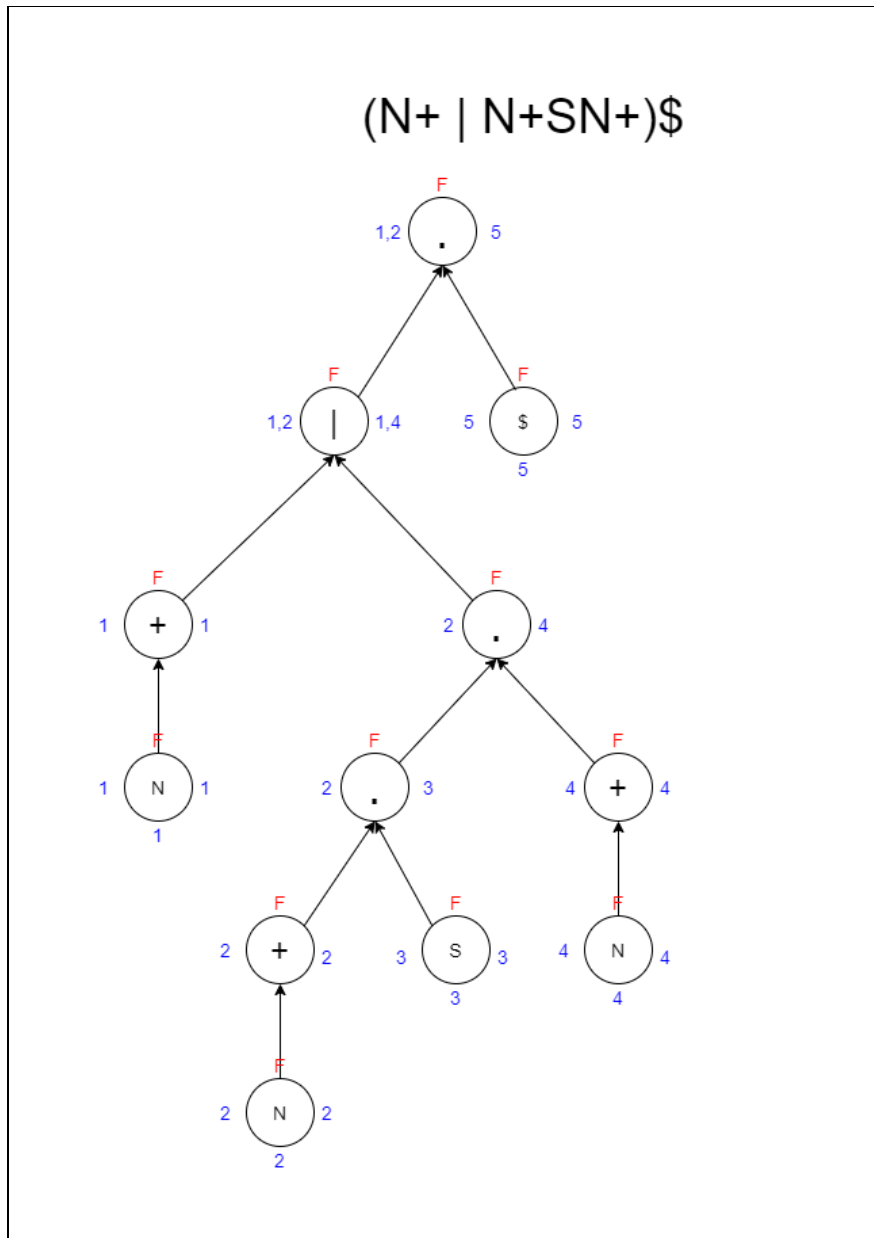
## Autómatas

**Alfabeto:**

$$\Sigma = \{L, N, S, C, A, D, P, U, E, R, F\}$$

Símbolo	Descripción	ASCII
L	Letras mayúsculas y minúsculas	65-90, 97-112, 209, 241
N	Cifras 0-9	48-57
S	Símbolos	33-126 (Exceptuando a los que están en esta tabla)
C	Comillas (“ ”)	34
A	Asterisco(*)	42
D	Diagonal(/)	47
P	Punto(.)	46
U	Guión bajo(_)	95
E	Espacio(“ ”)	32
R	Tecla Enter	13
F	Nueva Línea	10

## Números(Enteros/Decimal)



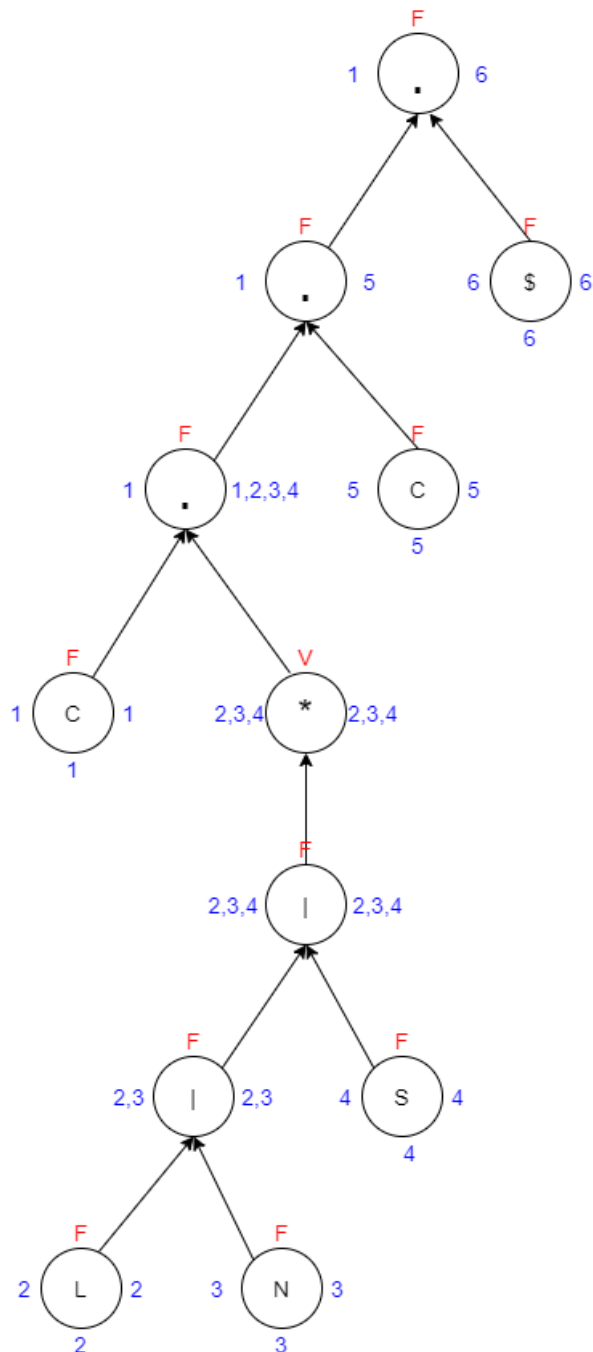
Números		
No	Sigma	Siguientes
1	N	1,5
2	N	2,3
3	P	4
4	N	4,5
5	\$	-

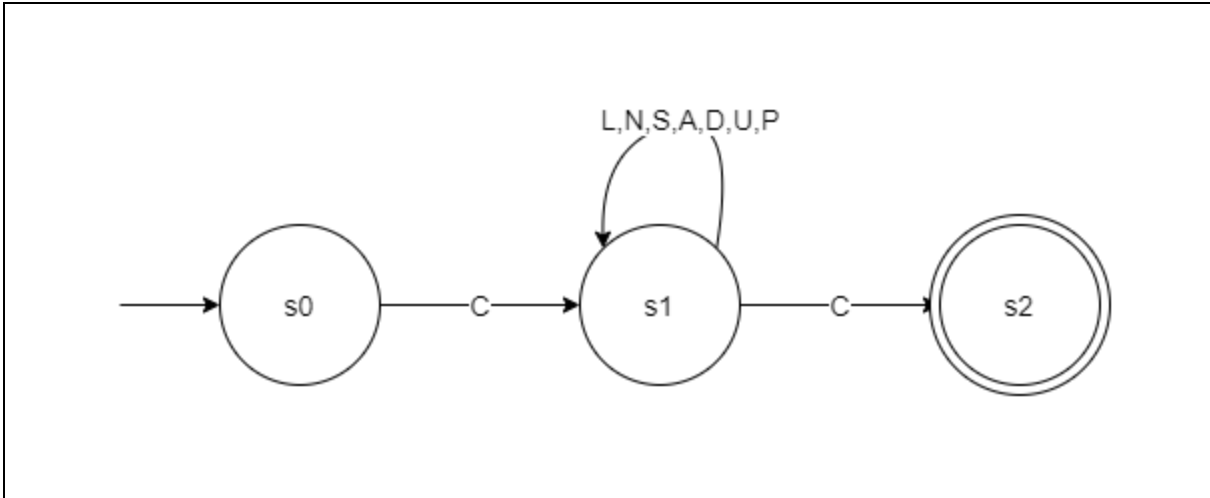
Estados	Siguientes	Transición
$s_0 = \{1,2\}$	$s(N) = s(1) \cup s(2) = \{1,2,3,5\} \rightarrow s_1$	$d(s_0, N) = s_1$
$s_1 = \{1,2,3,5\}$	$s(N) = s_1$	$d(s_1, N) = s_1$
	$s(P) = 4 \rightarrow s_2$	$d(s_1, P) = s_2$
$s_2 = \{4\}$	$s(N) = \{4,5\} \rightarrow s_3$	$d(s_2, N) = s_3$
$s_3 = \{4,5\}$	$s(N) = \{4,5\} \rightarrow s_3$	$d(s_3, N) = s_3$

	0	1
Estado	N	P(.)
0	1	
1	1	2
2	3	
3	3	

String(con comillas)

$C(L|N|S)^*C\$$





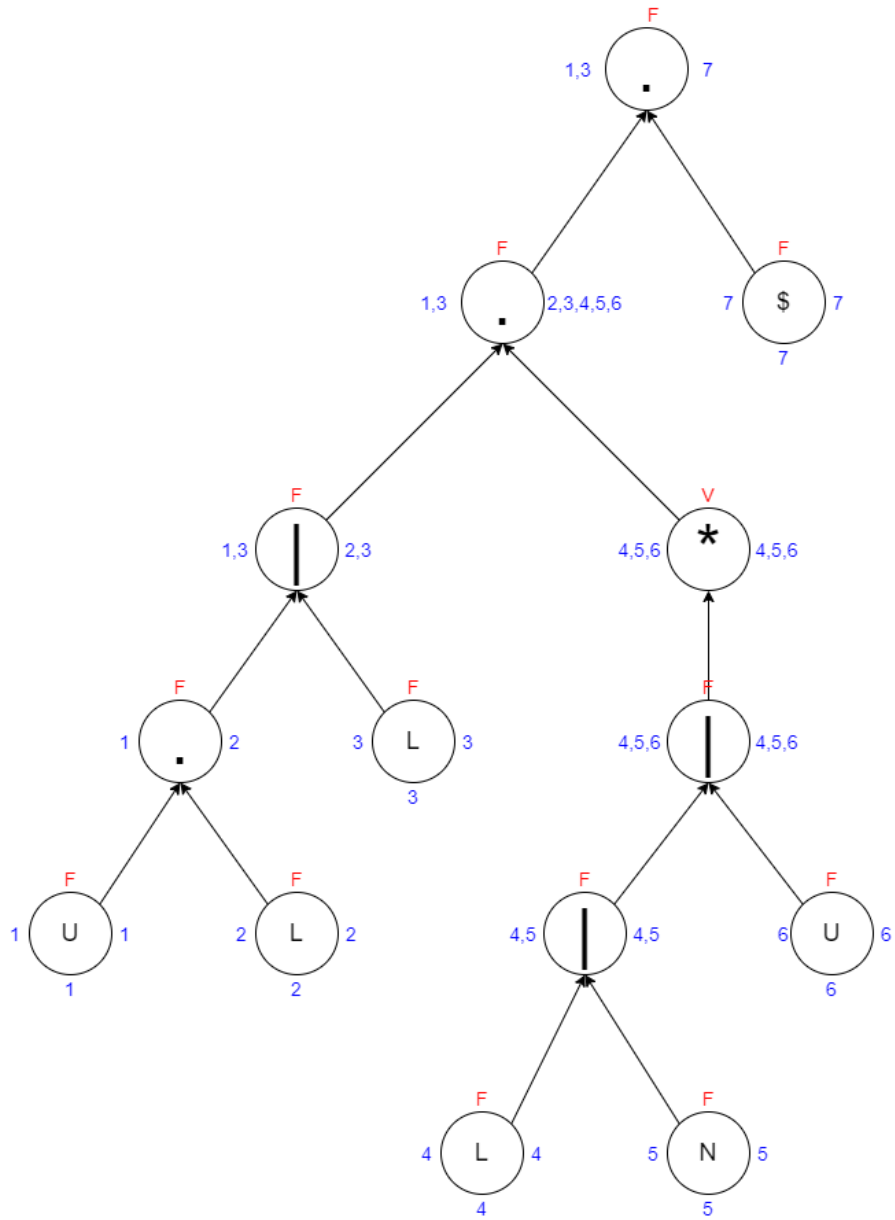
String con Comillas		
No	Sigma	Siguientes
1	C	2,3,4,5
2	L	2,3,4,5
3	N	2,3,4,5
4	S	2,3,4,5
5	C	6
6	\$	-

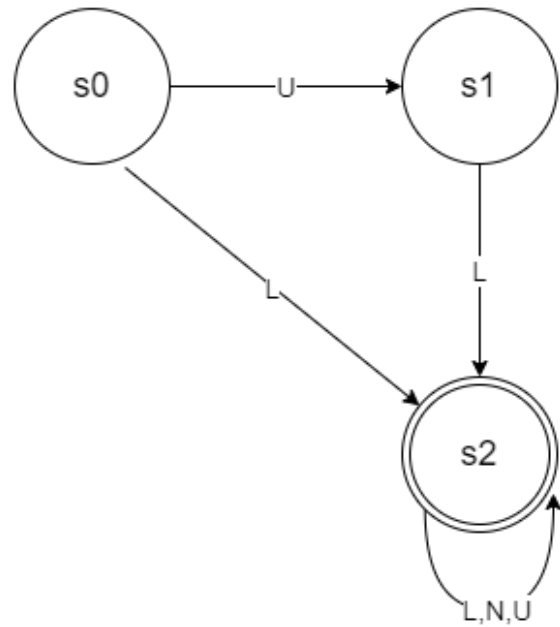
Estados	Siguientes	Transición
$s0 = \{1\}$	$s(C) = \{2,3,4,5\} \rightarrow s1$	$d(s0, C) = s1$
$s1 = \{2,3,4,5\}$	$s(L) = \{2,3,4,5\} \rightarrow s1$	$d(s1, L) = s1$
	$s(N) = s1$	$d(s1, N) = s1$
	$s(S) = s1$	$d(s1, S) = s1$
	$s(C) = \{6\} \rightarrow s2$	$d(s1, C) = s2$

ESTADO	C(")	L	N	S	A(*)	D(/)
0	1					
1	2	1	1	1	1	1
2						

## Cadenas / Identificadores

$(UL|L)(L|N|U)^*\$$





Cadenas		
No	Sigma	Siguientes
1	U	2
2	L	4,5,6,7
3	L	4,5,6,7
4	L	4,5,6,7
5	N	4,5,6,7
6	U	4,5,6,7
7	\$	-

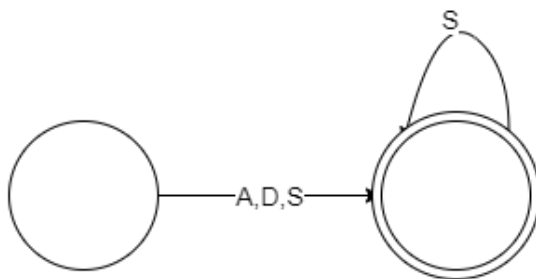
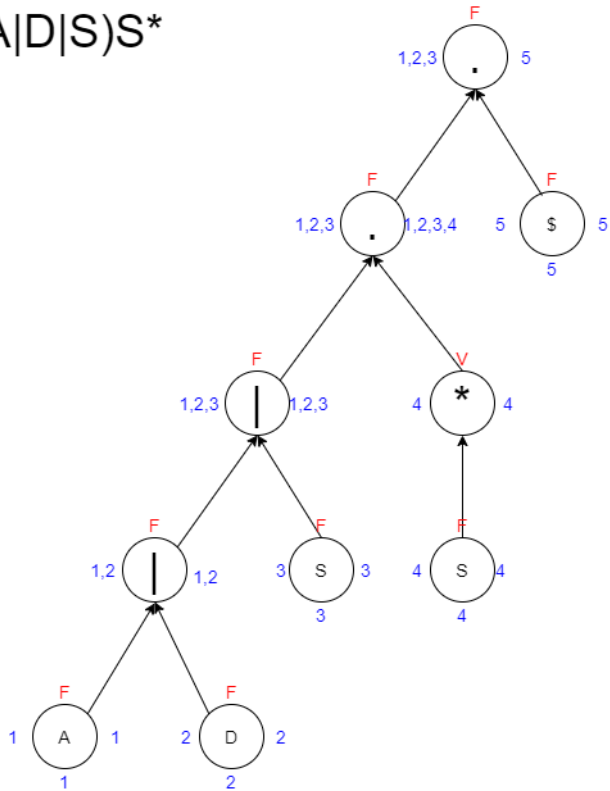
Estados	Siguientes	Transicion
$s_0 = \{1,3\}$	$s(U) = \{2\} \rightarrow s_1$	$d(s_0, U) = s_1$
	$s(L) = \{4,5,6,7\} \rightarrow s_2$	$d(s_0, L) = s_2$
$s_1 = \{2\}$	$s(L) = s_2$	$d(s_1, L) = s_2$
$s_2 = \{4,5,6,7\}$	$s(L) = s_2$	$d(s_2, L) = s_2$
	$s(N) = s_2$	$d(s_2, N) = s_2$
	$s(U) = s_2$	$d(s_2, U) = s_2$

	0	1	2
Estado	L	N	U
0	2		1
1	2		
2	2	2	2

## Comentario/Simbolo

$[(A|D|S)S^* \mid DD(L|N|S|C|A|D|P|U)^* \mid DA(L|N|S|C|A|D|P|U)^*AD]\$$

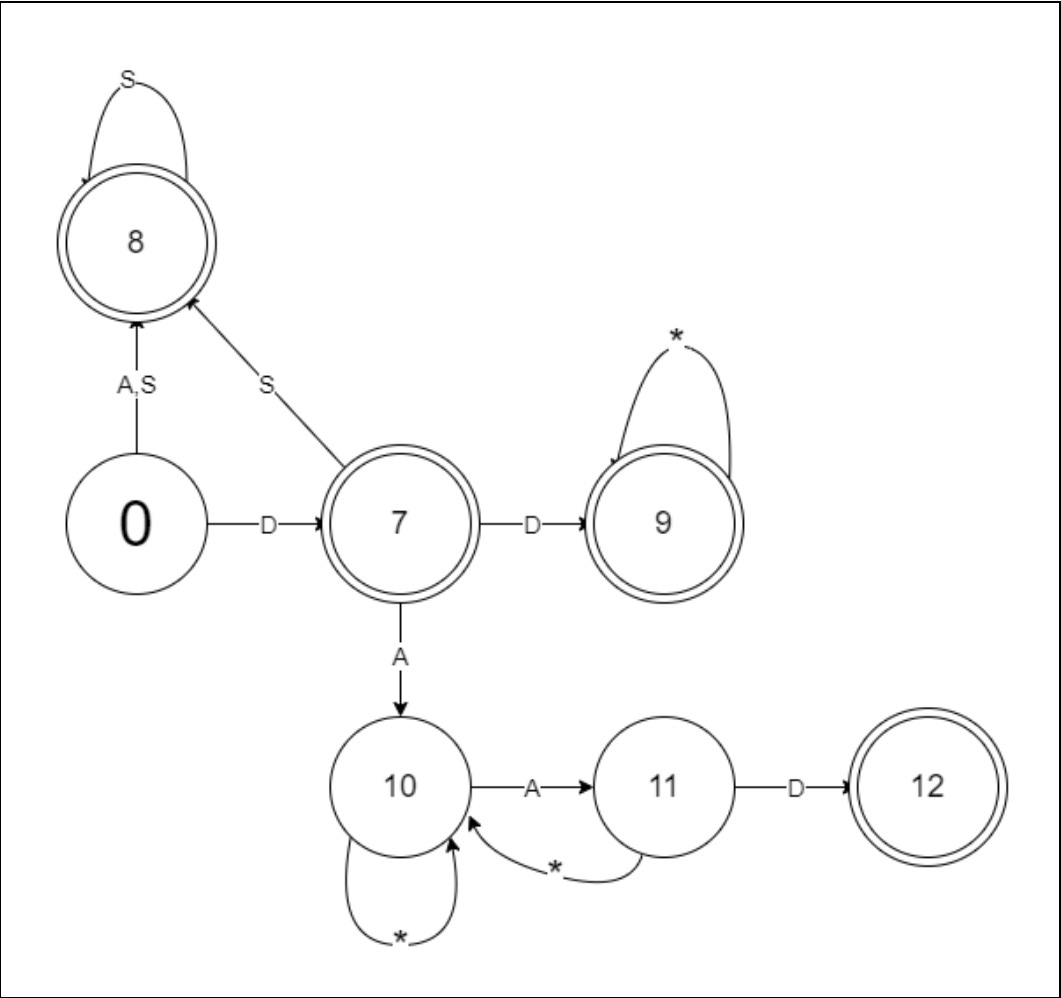
$(A|D|S)S^*$





Operadores		
No	Sigma	Siguientes
1	A	4,5
2	D	4,5
3	S	4,5
4	S	4,5
5	\$	-

Estados	Siguientes	Transicion
s0 = {1,2,3}	s(A) = {4,5} -> s1	d(s0,A) = s1
	s(D) = s1	d(s0,D) = s1
	s(S) = s1	d(s0,S) = s1
s1 = {4,5}	s(S) = s1	d(s1,S) = s1



## Tabla Final de Estados y Transiciones

[illegible]

## Gramática

<b>S</b>	-->	principal(){ <b>A</b> }
<b>A</b>	-->	<b>BA</b>
		<b>RA</b>
		<b>WA</b>
		<b>IA</b>
		<b>MA</b>
		<b>HA</b>
		<b>FA</b>
		<b>B'A</b>
		<b>E</b>
<b>B</b>	-->	"entero" id <b>EN</b> ;
		"decimal" id <b>DEC</b> ;
		"cadena" id <b>CAD</b> ;
		"booleano" id <b>BOL</b> ;
		"caracter" id <b>CAR</b> ;
<b>EN</b>	-->	<b>=XEN'</b>
		<b>EN'</b>
<b>EN'</b>	-->	, id <b>EN</b>
		<b>E</b>
<b>DEC</b>	-->	<b>=XDEC'</b>
		<b>DEC'</b>
<b>DEC'</b>	-->	, id <b>DEC</b>
		<b>E</b>
<b>CAD</b>	-->	<b>=CADENACAD'</b>

		<b>CAD'</b>
<b>CAD'</b>	-->	,id <b>CAD</b>
		<b>E</b>
<b>BOL</b>	-->	= <b>BOOLEANOBOL'</b>
		<b>BOL'</b>
<b>BOL'</b>	-->	,id <b>BOL</b>
		<b>E</b>
<b>CAR</b>	-->	= <b>CARACTERCAR'</b>
		<b>CAR'</b>
<b>CAR'</b>	-->	,id <b>CAR</b>
		<b>E</b>
<b>R</b>	-->	leer(id);
<b>W</b>	-->	imprimir( <b>K</b> );
<b>K</b>	-->	id <b>K'</b>
		<b>ENTEROK'</b>
		<b>DECIMALK'</b>
		<b>CADENAK'</b>
		<b>BOOLEANOK'</b>
		<b>CARACTERK'</b>
<b>K'</b>	-->	<b>+K</b>
		<b>E</b>
<b>B'</b>	-->	id <b>AS</b> ;
<b>AS</b>	-->	= <b>X</b>
		<b>+=X</b>
		<b>-=X</b>

		*= <b>X</b>
		/= <b>X</b>
		++
		--
<b>X</b>	-->	<b>TX'</b>
<b>X'</b>	-->	<b>+T</b>
		- <b>T</b>
		<b>E</b>
<b>T</b>	-->	<b>PT'</b>
<b>T'</b>	-->	<b>*P</b>
		/ <b>P</b>
		<b>E</b>
<b>P</b>	-->	<b>UP'</b>
<b>P'</b>	-->	<b>^P</b>
		<b>E</b>
<b>U</b>	-->	<b>-N</b>
		<b>N</b>
<b>N</b>	-->	ENTERO
		DECIMAL

		id
		(X)
I	-->	SI(C){A}I'
I'	-->	SINO_SI(C){A}I'
		SINO{A}
		E
C	-->	C'BOLEANOOL
		LOGOL
C'	-->	!C'
		E
LOG	-->	XCOM
COM	-->	OCX
OC	-->	==
		>=
		<=
		>
		<
		!=
OL	-->	&&C

		<b>C</b>
		<b>E</b>
<b>M</b>	-->	MIENTRAS( <b>C</b> ){ <b>A</b> }
<b>H</b>	-->	HACER{ <b>A</b> }MIENTRAS( <b>C</b> );
<b>F</b>	-->	DESDE <b>A'</b> HASTA <b>C</b> INCREMENTOENTERO{ <b>A</b> }
<b>A'</b>	-->	id=ENTERO
		enteroid=ENTERO

### Tabla de Primeros

Primeros	
No Terminal	Primeros
P(S)	{ principal }
P(A)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, E }
P(B)	{ entero, decimal, cadena, booleano, caracter }
P(EN)	{ =, ",", E }
P(EN')	{ ",", E }
P(DEC)	{ =, ",", E }
P(DEC')	{ ",", E }
P(CAD)	{ =, ",", E }
P(CAD')	{ ",", E }

P(BOL)	{ =, ",", E }
P(BOL')	{ ",", E }
P(CAR)	{ =, ",", E }
P(CAR')	{ ",", E }
P(R)	{ leer }
P(W)	{ imprimir }
P(K)	{ id, ENTERO, DECIMAL, CADENA, BOOLEANO, CARACTER }
P(K')	{ +, E }
P(B')	{ id }
P(AS)	{ =, +=, -=, *=, /=, ++, -- }
P(X)	{ -, ENTERO, DECIMAL, id, ( }
P(X')	{ +, -, E }
P(T)	{ -, ENTERO, DECIMAL, id, ( }
P(T')	{ *, /, E }
P(P)	{ -, ENTERO, DECIMAL, id, ( }
P(P')	{ ^, E }
P(U)	{ -, ENTERO, DECIMAL, id, ( }
P(N)	{ ENTERO, DECIMAL, id, ( }
P(I)	{ SI }
P(I')	{ SINO_SI, SINO, E }
P(C)	{ !, BOOLEANO, E, -, ENTERO, DECIMAL, id, ( }
P(C')	{ !, E }
P(LOG)	{ -, ENTERO, DECIMAL, id, ( }
P(COM)	{ ==, >=, <=, >, <, != }
P(OC)	{ ==, >=, <=, >, <, != }
P(OL)	{ &&,   , E }
P(M)	{ MIENTRAS }



P(H)	{ HACER }
P(F)	{ DESDE }
P(A')	{ id, entero }

### Tabla de Siguietes

Siguietes	
No Terminal	Siguietes
S(S)	{ \$ }
S(A)	{ ' ' }
S(B)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E }
S(EN)	{ ; }
S(EN')	{ ; }
S(DEC)	{ ; }
S(DEC')	{ ; }
S(CAD)	{ ; }
S(CAD')	{ ; }
S(BOL)	{ ; }
S(BOL')	{ ; }
S(CAR)	{ ; }
S(CAR')	{ ; }
S(R)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E }
S(W)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E, ==, >=, <=, >, <, !=, ), &&,   , E }

S(K)	{ }, ==, >=, <=, >, <, !=, &&,   , E }
S(K')	{ } }
S(B')	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, ==, >=, <=, >, <, !=, ), &&,   , E }
S(AS)	{ ;, ==, >=, <=, >, <, !=, ), &&,   , E }
S(X)	{ ',', ';', ==, >=, <=, >, <, !=, ), &&,   , INCREMENTO, E }
S(X')	{ ',', ';', ==, >=, <=, >, <, !=, ), &&,   , INCREMENTO, E }
S(T)	{ ',', ';', +, -, ==, >=, <=, >, <, !=, ), &&,   , INCREMENTO, E }
S(T')	{ ',', ';', +, -, INCREMENTO, E }
S(P)	{ ',', ';', +, -, *, /, INCREMENTO, E }
S(P')	{ ',', ';', +, -, *, /, INCREMENTO, E }
S(U)	{ ',', ';', +, -, *, /, ^, INCREMENTO, E }
S(N)	{ ',', ';', +, -, *, /, ^, INCREMENTO, E }
S(I)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, id, E, '}' }
S(I')	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, id, E, '}' }
S(C)	{ }, INCREMENTO }
S(C')	{ BOOLEANO }
S(LOG)	{ }, INCREMENTO, &&,   , E }
S(COM)	{ }, INCREMENTO, &&,   , E }
S(OC)	{ -, ENTERO, DECIMAL, id, ( }
S(OL)	{ }, INCREMENTO }
S(M)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E }
S(H)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E }
S(F)	{ entero, decimal, cadena, booleano, caracter, leer, imprimir, SI, MIENTRAS, HACER, DESDE, id, }, E }
S(A')	{ HASTA }

## **Tabla de Análisis Sintáctico**