

Tabla de Tokens

ID	Token	ER
0	Comentario 1	// &C*\n
1	Comentario 2	/*&C*/
2	PR int	int
3	PR double	double
4	PR char	char
5	PR bool	bool
6	PR String	string
7	Punto y coma	;
8	Igual	=
9	Identificador	L(L D _)*
10	Número entero	D+
11	Decimal	D+ [.D+]?
12	Suma	+
13	Resta	-
14	División	/
15	Multiplicación	*
16	PR void	void
17	Paréntesis abierto	(
18	Paréntesis cerrado)
19	Llave abierta	{
20	Llave cerrada	}
21	PR def	def
22	PR main	main
23	Coma	,
25	PR if	if
26	PR else	else

27	PR switch	switch
28	PR case	case
29	PR break	break
30	Dos puntos	:
31	PR default	default
32	PR for	for
33	Incremento	++
34	Decremento	--
35	PR while	while
36	PR do	do
37	PR Console	Console
38	Punto	.
39	PR Write	Write
40	Cadena	"&C*"
41	PR true	true
42	PR false	false
43	Carácter	'L'
44	PR return	return
45	PR continue	continue
46	Condición Y	&&
47	Condición O	
48	Condición not	!
49	Mayor	>
50	Menor	<
51	Menor o igual	<=
52	Mayor o igual	>=
53	igual	==
54	Distinto	!=
55	html	'&C+'

Gramática

PROGRAMA -> LISTASENTECIAS

PROGRAMA -> MAIN

LISTASENTECIAS -> SENTENCIAS

LISTASENTECIAS -> LISTASENTECIAS SENTENCIAS

LISTASENTECIAS -> FUNCIONES

LISTASENTECIAS -> LISTASENTECIAS FUNCIONES

FUNCIONES -> METODO

FUNCIONES -> FUNCION

SENTENCIAS -> WHILE SENTENCIAS

SENTENCIAS -> DO_WHILE SENTENCIAS

SENTENCIAS -> FOR SENTENCIAS

SENTENCIAS -> SWITCH SENTENCIAS

SENTENCIAS -> VARIABLES SENTENCIAS

SENTENCIAS -> BREAK

SENTENCIAS -> CONTINUE

SENTENCIAS -> LLAMADA ;

METODO -> void id (PARAMETROS) { SENTENCIAS }

FUNCION -> TIPO id (PARAMETROS) { SENTENCIAS RETURN }

RETURN-> return DATO ;

RETURN-> ERROR ;

MAIN -> void main () { LISTASENTECIAS }

TIPO -> string

TIPO -> char

TIPO -> int

TIPO -> double

NUMEROS -> NUMEROS + N1

NUMEROS -> NUMEROS - N1

NUMEROS -> N1

N1 -> N1 * N2

N1 -> N1 / N2

N1 -> N2

N2 -> (N3)

N2 -> N3

N3 -> -N4

N3 -> N4

N4 -> numero

N4 -> double

N4 -> id

N4 -> LLAMADA

LLAMADA -> id (PENTRADA)

LLAMADA -> ERROR ;

PENTRADA -> DATO , DATO

PENTRADA -> DATO

PARAMETROS -> T P1

P1 -> id P2

P2 -> , PARAMETROS| epsilon

DATO -> identificador

DATO -> caracter

DATO -> true

DATO -> false

DATO -> NUMEROS

DATO -> LLAMADA

COMPARADOR -> DATO > DATO

COMPARADOR -> DATO < DATO

COMPARADOR -> DATO >= DATO

COMPARADOR -> DATO <= DATO

COMPARADOR -> DATO == DATO

COMPARADOR -> DATO != DATO

CONDICION -> CONDICION || C1

CONDICION -> C1

C1 -> C1 && C2

C1 -> C2

C2 -> ! COMPARADOR

C2 -> COMPARADOR

BREAK -> break ;

BREAK -> ERROR ;

CONTINUE -> continue ;

CONTINUE -> ERROR ;

IMPRESION -> console . write (I0) ;

IMPRESION -> ERROR ;

I0 -> I0 + I0

I0 -> cadena

I0 -> numero

I0 -> id

I0 -> decimal

VARIABLES -> TIPO V10

VARIABLES -> V10

V10 -> V1 ;

V10 -> ERROR ;

V1 -> V2 , V2

V1 -> V2

V2 -> id V3

V3 -> = DATO

IF -> if (CONDICION) { SENTENCIAS } lprim

lprim -> else if (CONDICION) { SENTENCIAS } lprim

lprim -> lprim2

lprim2-> else { SENTENCIAS }

lprim2-> epsilon

SWITCH -> switch (id) { S0 }

S0 -> case DATO : sentencias S1

S1 -> BREAK S2

S1 -> S2

S2 -> S0

S2 -> default: sentencia BREAK

S2 -> epsilon

FOR -> for (VARIABLES ; CONDICION ; CAMBIO) { SENTENCIAS }

CAMBIO -> ++ id

CAMBIO -> -- id

CAMBIO -> id --

CAMBIO -> id ++

CAMBIO -> epsilon

CAMBIO -> V1

WHILE -> while (CONDICION) { SENTENCIAS }

DO_While -> do { SENTENCIAS } while (CONDICION)