Tabla LL

NT / T	if	let	id	input	alert	eof	(	;	)	,	{	:	}	entero	number	boolean	function	string	return	cadena	switch	case	default	break	=	||	|=	==	+	-	/	$
E			E -> R 1				E -> R 1							E -> R 1						E -> R 1												
1								1 -> lambda	1 -> lambda	1 -> lambda																1 -> || R 1						
R			R -> U 2				R -> U 2							R -> U 2						R -> U 2												
2								2 -> lambda	2 -> lambda	2 -> lambda																		2 -> == U 2				
U			U -> V 3				U -> V 3							U -> V 3						U -> V 3												
3								3 -> lambda	3 -> lambda	3 -> lambda																		3 -> lambda	3 -> G V 3	3 -> G V 3	3 -> lambda	
G																													G -> +	G -> -		
V			V -> Z 4				V -> Z 4							V -> Z 4						V -> Z 4												
4								4 -> lambda	4 -> lambda	4 -> lambda																		4 -> lambda	4 -> lambda	4 -> lambda	4 -> / Z 4	
Z			Z -> id O				Z -> (E)							Z -> entero						Z -> cadena												
O							O -> (L)	O -> lambda	O -> lambda	O -> lambda																		O -> lambda	O -> lambda	O -> lambda	O -> lambda	
S			S -> id D ;	S -> input (id) ;	S -> alert (E) ;														S -> return X ;													
D							D -> (L)																		D -> = E		D -> |= E					
L			L -> E Q				L -> E Q							L -> E Q						L -> E Q												
Q									Q -> lambda	Q -> , E Q																						
X			X -> E				X -> E	X -> lambda						X -> E						X -> E												
B	B -> if (E) S	B -> let T id ;	B -> S	B -> S	B -> S														B -> S		B -> switch (E) { Y }											
Y																						Y -> case entero : C M										
M																						M -> N	M -> N	M -> break ; N								
N													N -> lambda									N -> Y	N -> default : C									
F																	F -> function id (A) { C }															
T															T -> number	T -> boolean		T -> string														
H			H -> lambda												H -> T	H -> T		H -> T														
A									A -> lambda						A -> T id K	A -> T id K		A -> T id K														
K									K -> lambda	K -> T id K																						
C	C -> B C	C -> B C	C -> B C	C -> B C	C -> B C								C -> lambda						C -> B C		C -> B C	C -> lambda	C -> lambda	C -> lambda								
P	P -> B P	P -> B P	P -> B P	P -> B P	P -> B P	P -> eof											P -> F P		P -> B P		P -> B P											P -> lambda