

Gramática libre de contexto:

G = {N, T, S, P}

S = {E}

N = {E, C, V, D, T, L}

T = {Parametro, NUM, DIGITO, OPERADOR}

P =

E	-->	funcion principal { C }
		funcion id (PARAMETRO) { C }
C	-->	V ;
		V = T L ;
		imprimir ();
		while() { C }
		if () { C }
		for (V = NUM ; id < NUM ; ID++) { C }
		id (id) ;
		/*comentario/*
V	-->	variable D id
D	-->	entero
		decimal
		booleano
		cadena
		caracter
T	-->	NUM
		NUM . NUM
		true
		false
		"cadena"
L	-->	id OPERADOR id
		e
PARAMETRO	-->	D id
		e
NUM	-->	DIGITO DIGITO
		e
DIGITO	-->	0 1 2 3 4 5 6 7 8 9
		e
OPERADOR	-->	+ - * / % = == < > >= <=
		e

Gramática LL1:

Tabla de primeros:

No.Terminal	Primeros
E	funcion principal, ID
C	Variable, imprimir, while, if, for, id, comentario
V	Variable
D	entero, decimal, booleano, carácter, cadena
T	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, true, false, cadena
L	Id, e
Parametro	entero, decimal, booleano, carácter, cadena, e
Num	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Digito	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Operador	+ - * / % = == < > >= <=

Tabla de siguientes:

No.Terminal	Siguientes
E	\$
C	\$
V	;,
D	id
T	id, e, \$
L	;
Parametro	variable, imprimir, while
Num	;, id, e, \$
DIGITO	;, id, e, \$
OPERADOR	id

Tabla sintáctica:

	funcion principal	id	variable	imprimir	while	if
E	funcion principal { C }	funcion ID (PARAMETRO) { C }				
C		for (V = NUM ; ID < NUM ; ID++) { C }	V ;	V = T L ;	imprimir ();	while() { C }
V			variable			
D			D ID			
T						
L		ID OPERADOR ID				
Parametro			e	e	e	
Num						
Digito						
Operador						

for	entero	decimal	booleano	carácter	cadena	true
if () { C }						
	entero	decimal	booleano	carácter	cadena	
					false	NUM.NUM
	D ID	e	e	e	e	

false	1, 2, 3,4,5,6,7,8,9	+ - * / % = == < > > <=	;	\$
true	NUM			
			e	
	DIGITO DIGITO			
	0 1 2 3 4 5 6 7 8 9			
		+ - * / % = == < > > <=		