Gramática

A → Principal {B

B → L}

L → DL | GL | FL | ZL | OL | JL | ε

D → D’

D’ → Int ID Q | Double ID W | Boolean ID R | String ID T | Char ID Y

Q → =ENTERO | , I | ε

W → =DECIMAL | , I | ε

R → =BOOLEANO | , I | ε

T → =CADENA O’ | , I | ε

Y → =CARACTER | , I | ε

I → ID I

I’ → , I | ε

G → escribir (C | leer (ID);

C → M);

M → ENTERO M’ | DECIMAL M’ | CADENA M’ | ID M’

M’ → + M | ε

C’ → ID);

S’ → {L}

F → Si(V) S’ E E’ | Mientras (V) S’ | Hacer S’ Mientras (V) | Desde ID=ENTERO HASTA ID B ENTERO INCREMENTO ENTERO S’

V → P V’ X W’ P’)

V’ → ENTERO | DECIMAL | BOOLEANO | CADENA | ID

X → = | < | > | = | & | | | == | <= | >= | ¡=

P → (| ε

P’ →) Q’ | ε

X’ → && V | || V

B → < | > | = | <= | >=

E → Sino si (V)S’ E | ε

E’ → Sino S’ | ε

Z → ID J;

Z’ → O | ENTERO | DECIMAL | BOOLEANO | CADENA | CARACTER

O → ID= ENTERO N | DECIMAL N | ID N | (N)

O’ → +N’ | ε

N → + O | -’ | / O | * O | ε

N’ → CADENA O’ | ID O’

J → ++ | -- | = Z’

Cálculo de primeros

Terminal	**Primeros**
P(A)	{Principal}
P(B)	{entero,decimal.booleano,string,char, escribir, leer. Si, mientras, hacer, desde, id}
P(L)	{entero,decimal.booleano,string,char, escribir, leer. Si, mientras, hacer, desde, id, ε}
P(Z)	{id}
P(Z’)	{entero, decimal, booleano, cadena, carácter}
P(D)	{entero, decimal, booleano, cadena, carácter}
P(D’)	{entero, decimal, booleano, cadena, carácter}
P(Q)	{ = , , , ε }

P(W)	{ = , , , Ɛ }
P(R)	{ = , , , Ɛ }
P(T)	{ = , , , Ɛ }
P(Y)	{ = , , , Ɛ }
P(I)	{ID}
P(I’)	{ , , Ɛ}
P(G)	{escribir, leer}
P(M)	{entero, decimal, cadena, id}
P(M’)	{+, Ɛ)
P(C)	{ cadena, entero, decimal, id }
P(F)	{si, mientras, hacer ,desde}
P(V)	{(,id,entero, decimal, booleano, cadena}
P(V’)	{id,entero, decimal, booleano, cadena}
P(X)	{ >, <, ==, <=, >=, != }
P(P’)	{ SINO_SI, Ɛ }
P(Q’)	{ &&, ‖}
P(B)	{ &&, ‖}
P(E)	{ SINO_SI, Ɛ }
P(E’)	{ SINO, Ɛ }
P(O)	{ ID}
P(O’’)	{+,-,/,*, Ɛ)
P(J)	{ID}
P(J’’)	{ID}

Cálculo de siguientes

Terminal	**Siguientes**
S(A)	{$}
S(B)	{ $ }
S(L)	{ } }
S(Z)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(Z’)	{ ; }
S(D)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(D’)	{ ; }
S(Q)	{ ; }
S(W)	{ ; }
S(R)	{ ; }
S(T)	{ ; }
S(Y)	{ ; }
S(I)	{ ; }
S(I’)	{ ; }
S(G)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(M)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(M’)	{), ;)}
S(C)	{), ;)}
S(F)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(V)	{) }
S(V’)	{ >, <, ==, <=, >=, != }
S(X)	{entero, decimal, booleano, cadena, carácter, id}
S(P’)	{) }
S(Q’)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id, Ɛ}
S(B)	{entero}
S(E)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, sino, mientras, hacer, desde, id, Ɛ}
S(E’)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(O)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(O’)	{ ;)}
S(O’’)	{ ;)}
S(J)	{entero, decimal, booleano, cadena, carácter, escribir, leer, si, mientras, hacer, desde, id}
S(J’)	{ ; }