Gramatica

```
prg : declarations
      statements

declarations : decl
             : declaration decl

decl : declV
     | declF
     | declL

declV : VAR ids : tipo ENTER

ids : ID
    | ids , ID

tipo: ENTERO
    | REAL
    | BOOCEANO

declL : LISTA ID = lista ENTER

lista: [ ]
     | [elementos]

elementos: elemento
         | elementos , elemento

elemento : INT
         | FLOAT
         | BOOLEANO
         | lista
         | STRING
         | ID

declF: FUNCION ID ( parametros ) : ENTERO ENTER
       declarationsF statementsI DEVUELVE return
     | FUNCION ID (       ) : ENTERO ENTER
       declarationsF statementsI DEVUELVE return

parametros: ID : tipo
          | parametros , ID : tipo

declarationsF: TAB decl
             | declarationsF TAB decl

return : ID ENTER
       : INT ENTER

StatementI: TAB stat
          : StatementsI TAB stat

Statements: stat
          : Statements stat

stat: atrib ENTER
    | condition
    | ciclo

atrib : ID = exp

exp : exp + termo
    | exp - termo
    | termo

termo: termo * factor
     | termo / factor
     | termo ^ factor
     | factor

factor : INT
       | FLOAT
       | ID
       | ( exp )
```

Conditions : SI condition ENTONCES ENTER
STATEMENTSI
| SI expL ENTONCES ENTER
STATEMENTSI SI NO ENTER
STATEMENTSI

expL : expL o termoB
| termoB

termoB : termoB y factorB
| factorB

factorB : condition
| BOOLEANO
| (expL)

condition : exp op exp

op : >
| <
| IGUAL
| DIFERENT
| > IGUAL
| < IGUAL

exp : exp + termo
| exp - termo
| termo

termo : termo * factor
| termo / factor
| termo ^ factor
| factor

factor : INT
| FLOAT
| ID
| (exp)

Ciclos : ENCUANTO expL HACER ENTER
statementsI
| PARA expL SIGUIENTE atrib. ENTER
| PARA expL SIGUIENTE atrib HACER ENTER
statementsI