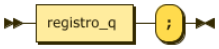
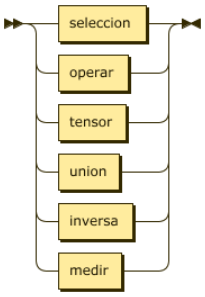
prog:




```
prog     ::= registro_q ';'
```

no references

expr:

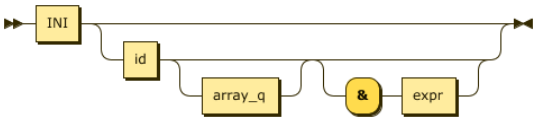



```
expr     ::= seleccion
           | operar
           | tensor
           | union
           | inversa
           | medir
```

referenced by:

- inversa
- medir
- operar
- registro_q
- seleccion
- tensor
- union

registro_q:

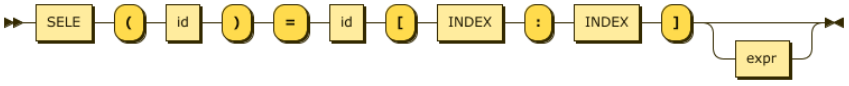



```
registro_q
         ::= INI ( id array_q? ( '&' expr )? )?
```

referenced by:

- prog

seleccion:

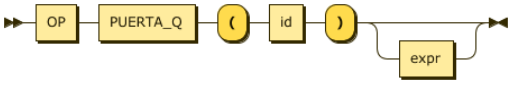



```
seleccion
         ::= SELE '(' id ')' '=' id '[' INDEX ':' INDEX ']' expr?
```

referenced by:

- expr

operar:

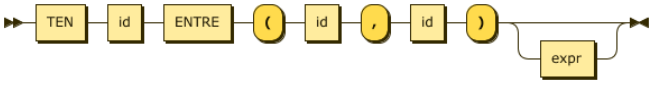



```
operar   ::= OP PUERTA_Q '(' id ')' expr?
```

referenced by:

- expr

tensor:

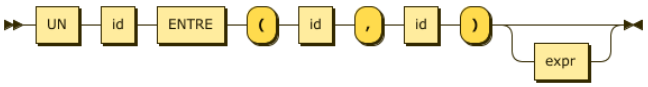



```
tensor   ::= TEN id ENTRE '(' id ',' id ')' expr?
```

referenced by:

- expr

union:

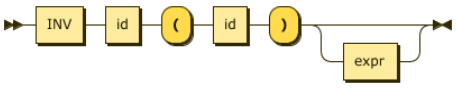



```
union    ::= UN id ENTRE '(' id ',' id ')' expr?
```

referenced by:

- expr

inversa:

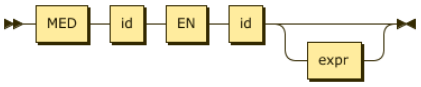



```
inversa  ::= INV id '(' id ')' expr?
```

referenced by:

- expr

medir:

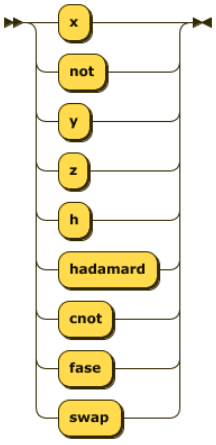



```
medir    ::= MED id EN id expr?
```

referenced by:

- expr

PUERTA_Q:




```
PUERTA_Q ::= 'x'
           | 'not'
           | 'y'
           | 'z'
           | 'h'
           | 'hadamard'
           | 'cnot'
           | 'fase'
           | 'swap'
```

referenced by:

- operar

INI:


```
INI      ::= 'inicio'
           | 'INICIO'
```

referenced by:

- registro_q

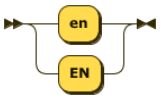
SELE:


```
SELE     ::= 'seleccion'
           | 'SELECCION'
```

referenced by:

- seleccion

EN:

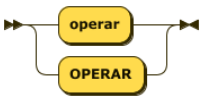



```
EN       ::= 'en'
           | 'EN'
```

referenced by:

- medir

OP:

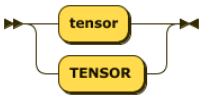



```
OP       ::= 'operar'
           | 'OPERAR'
```

referenced by:

- operar

TEN:

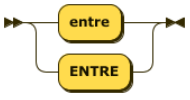



```
TEN      ::= 'tensor'
           | 'TENSOR'
```

referenced by:

- tensor

ENTRE:

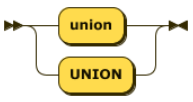



```
ENTRE    ::= 'entre'
           | 'ENTRE'
```

referenced by:

- tensor
- union

UN:

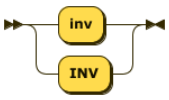



```
UN       ::= 'union'
           | 'UNION'
```

referenced by:

- union

INV:

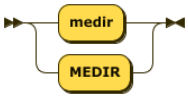



```
INV      ::= 'inv'
           | 'INV'
```

referenced by:

- inversa

MED:

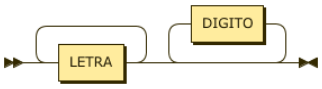



```
MED      ::= 'medir'
           | 'MEDIR'
```

referenced by:

- medir

id:

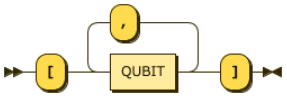



```
id       ::= LETRA+ DIGITO*
```

referenced by:

- array
- inversa
- medir
- operar
- registro_q
- seleccion
- tensor
- union
- variable
- variable_declar

array_q:

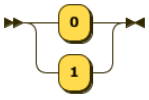



```
array_q  ::= '[' QUBIT ( ',' QUBIT )* ']'
```

referenced by:

- registro_q

QUBIT:

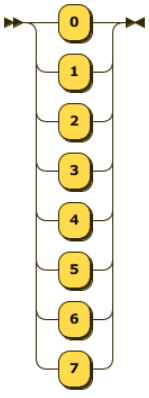



```
QUBIT    ::= '0'
           | '1'
```

referenced by:

- array_q

INDEX:



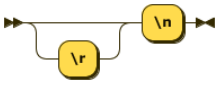

```
INDEX    ::= '0'
           | '1'
           | '2'
           | '3'
           | '4'
           | '5'
```

```
          | '6'
          | '7'
```

referenced by:

- seleccion

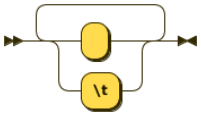
NEWLINE:




```
NEWLINE  ::= '\r'? '\n'
```

no references

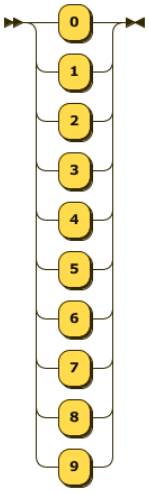
WHITESPACE:




```
WHITESPACE
         ::= ( ' ' | '\t' )+
```

no references

DIGITO:

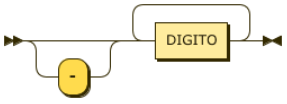



```
DIGITO   ::= '0'
           | '1'
           | '2'
           | '3'
           | '4'
           | '5'
           | '6'
           | '7'
           | '8'
           | '9'
```

referenced by:

- N_FLOAT
- N_INTEGER
- id

N_INTEGER:

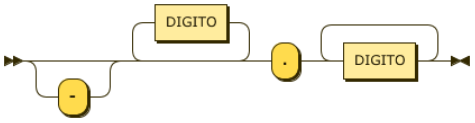



```
N_INTEGER
         ::= '-'? DIGITO+
```

referenced by:

- array
- variable_declar

N_FLOAT:

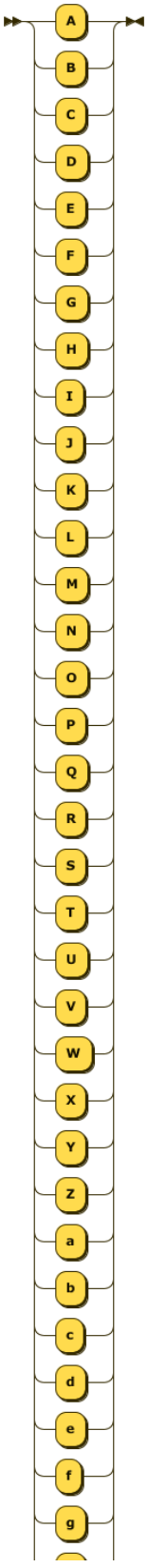


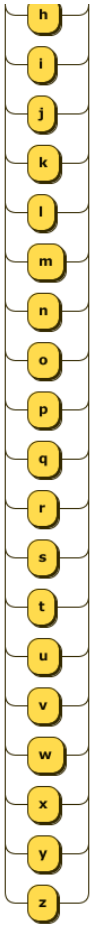

```
N_FLOAT  ::= '-'? DIGITO* '.' DIGITO+
```

referenced by:

- array
- variable_declar

LETRA:



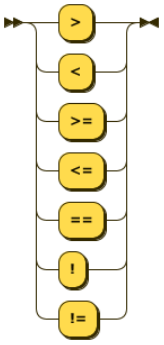



```
LETRA    ::= 'A'
           | 'B'
           | 'C'
           | 'D'
           | 'E'
           | 'F'
           | 'G'
           | 'H'
           | 'I'
           | 'J'
           | 'K'
           | 'L'
           | 'M'
           | 'N'
           | 'O'
           | 'P'
           | 'Q'
           | 'R'
           | 'S'
           | 'T'
           | 'U'
           | 'V'
           | 'W'
           | 'X'
           | 'Y'
           | 'Z'
           | 'a'
           | 'b'
           | 'c'
           | 'd'
           | 'e'
           | 'f'
           | 'g'
           | 'h'
           | 'i'
           | 'j'
           | 'k'
           | 'l'
           | 'm'
           | 'n'
           | 'o'
           | 'p'
           | 'q'
           | 'r'
           | 's'
           | 't'
           | 'u'
           | 'v'
           | 'w'
           | 'x'
           | 'y'
           | 'z'
```

referenced by:

- id

OP_LOGICA:

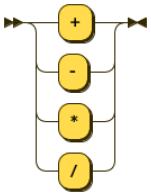



```
OP_LOGICA
         ::= '>'
           | '<'
           | '>='
           | '<='
           | '=='
           | '!'
           | '!='
```

referenced by:

- condicion

OP_ARITMETICA:

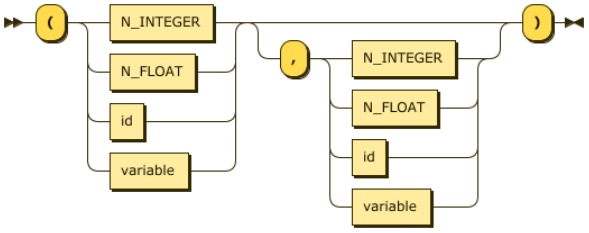



```
OP_ARITMETICA
         ::= '+'
           | '-'
           | '*'
           | '/'
```

referenced by:

- variable_declar

array:

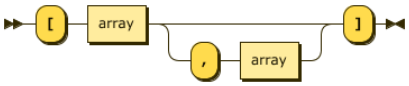



```
array    ::= '(' ( N_INTEGER | N_FLOAT | id | variable ) ( ',' ( N_INTEGER | N_FLOAT | id | variable ) )? ')'
```

referenced by:

- matriz
- variable_declar

matriz:

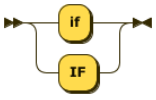



```
matriz   ::= '[' array ( ',' array )? ']'
```

referenced by:

- variable_declar

IF:

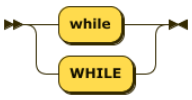



```
IF       ::= 'if'
           | 'IF'
```

referenced by:

- if_sentencia

WHILE:

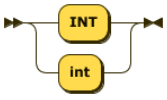



```
WHILE    ::= 'while'
           | 'WHILE'
```

referenced by:

- while_sentencia

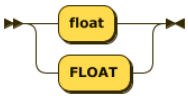
INT:




```
INT      ::= 'INT'
           | 'int'
```

no references

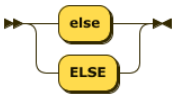
FLOAT:




```
FLOAT    ::= 'float'
           | 'FLOAT'
```

no references

ELSE:

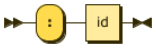



```
ELSE     ::= 'else'
           | 'ELSE'
```

referenced by:

- else_sentencia

variable:

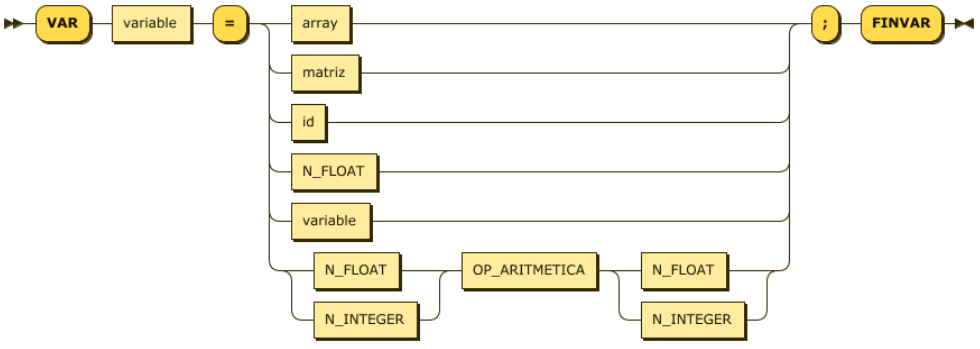



```
variable ::= ':' id
```

referenced by:

- array
- condicion
- variable_declar

variable_declar:




```
variable_declar
        ::= 'VAR' variable '=' ( array | matriz | id | N_FLOAT | variable | ( N_FLOAT | N_INTEGER ) OP_ARITMETICA ( N_FLOAT | N_INTEGER ) ) ';'
            'FINVAR'
```

referenced by:

- sentencia

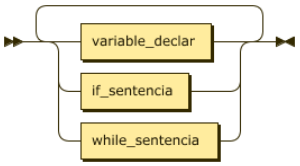
condicion:


```
condicion
        ::= variable OP_LOGICA variable
```

referenced by:

- if_sentencia
- while_sentencia

sentencia:

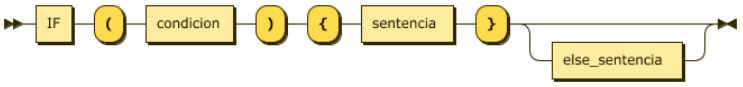



```
sentencia
        ::= ( variable_declar | if_sentencia | while_sentencia )+
```

referenced by:

- else_sentencia
- if_sentencia
- while_sentencia

if_sentencia:

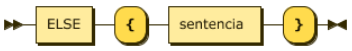



```
if_sentencia
        ::= IF '(' condicion ')' '{' sentencia '}' else_sentencia?
```

referenced by:

- sentencia

else_sentencia:




```
else_sentencia
        ::= ELSE '{' sentencia '}'
```

referenced by:

- if_sentencia

while_sentencia:


```
while_sentencia
        ::= WHILE '(' condicion ')' '{' sentencia '}'
```

referenced by:

- sentencia
