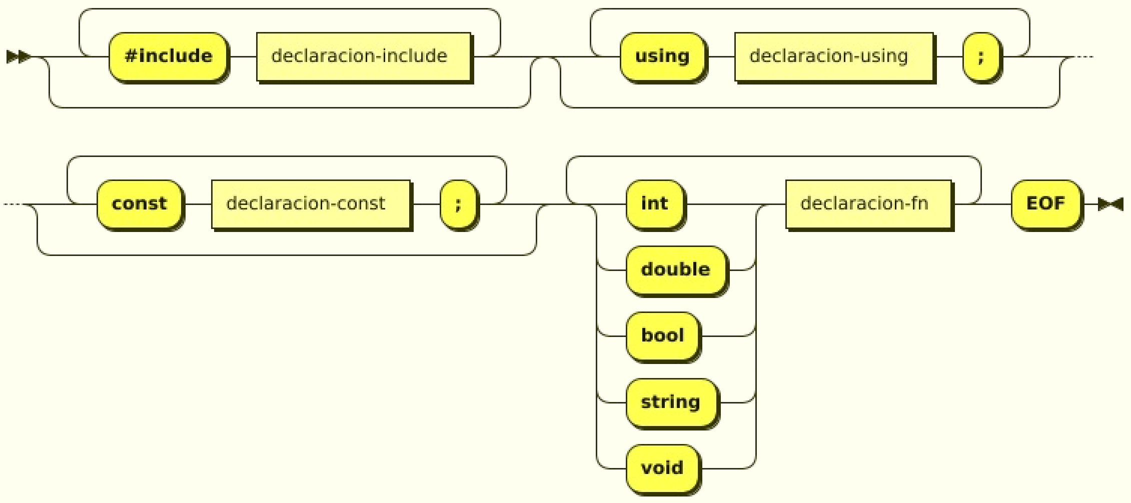
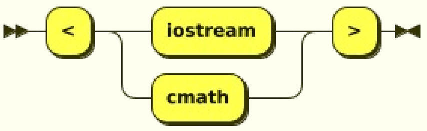
programa:



```
programa ::= ( '#include' declaracion-include )* ( 'using' declaracion-using ';' )* ( 'const' declaracion-const ';' )* ( ( 'int' |
           'double' | 'bool' | 'string' | 'void' ) declaracion-fn )+ 'EOF'
```

no references

declaracion-include:

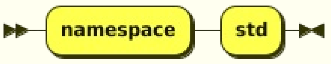


```
declaracion-include
         ::= '<' ( 'iostream' | 'cmath' ) '>'
```

referenced by:

- programa

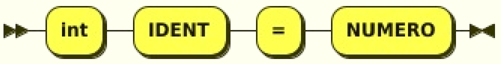
declaracion-using-namespace:



```
declaracion-using-namespace
         ::= 'namespace' 'std'
```

no references

declaracion-const:

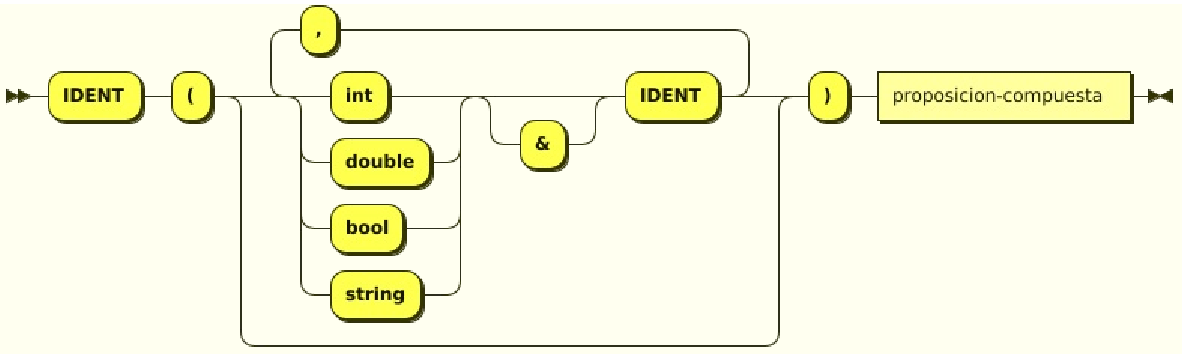


```
declaracion-const
         ::= 'int' 'IDENT' '=' 'NUMERO'
```

referenced by:

- programa

declaracion-fn:

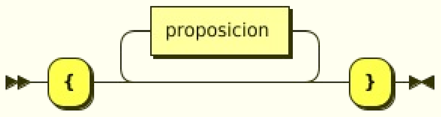


```
declaracion-fn
         ::= 'IDENT' '(' ( ( 'int' | 'double' | 'bool' | 'string' ) '&'? 'IDENT' ( ',' ( 'int' | 'double' | 'bool' | 'string' ) '&'?
             'IDENT' )* )? ')' proposicion-compuesta
```

referenced by:

- programa

proposicion-compuesta:

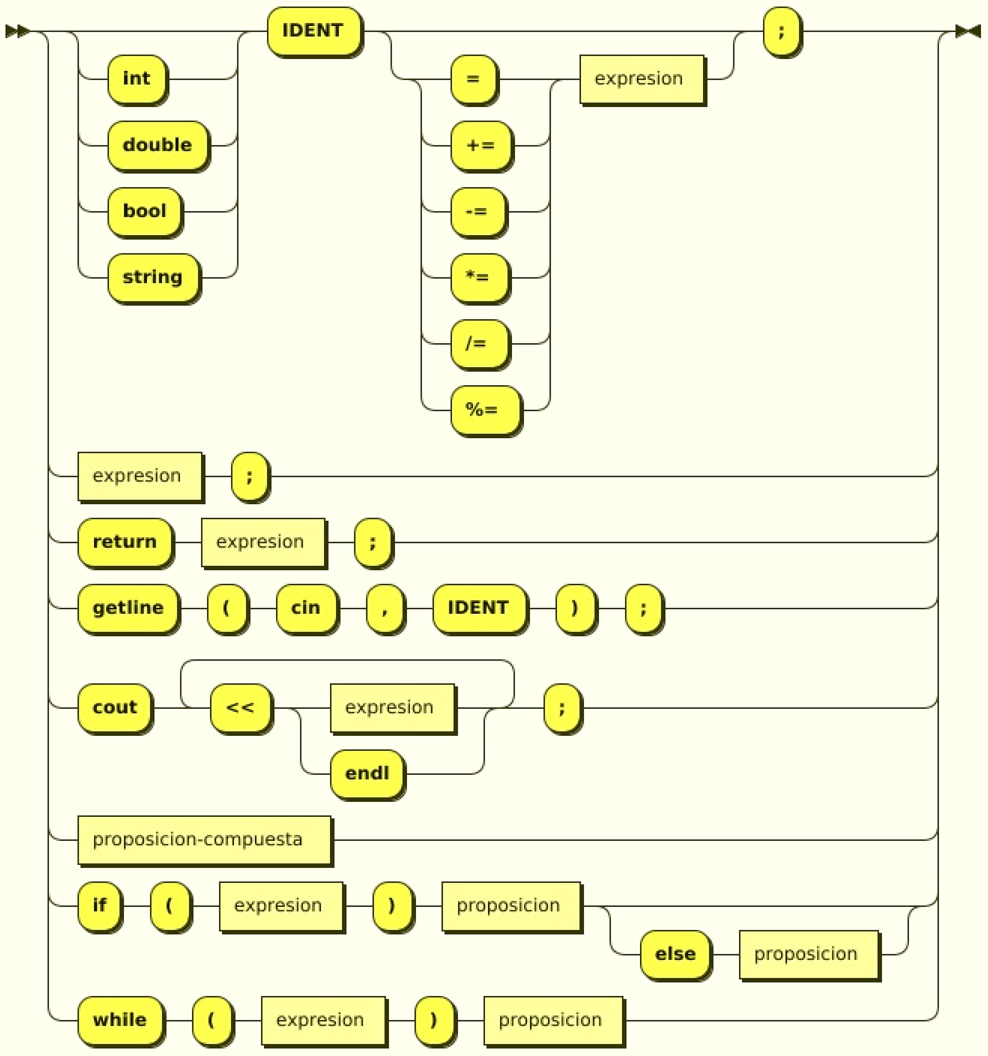


```
proposicion-compuesta
         ::= '{' proposicion* '}'
```

referenced by:

- declaracion-fn
- proposicion

proposicion:



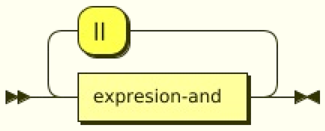
```
proposicion
```

```
         ::= ( 'int' | 'double' | 'bool' | 'string' )? 'IDENT' ( ( '=' | '+=' | '-=' | '*=' | '/=' | '%=' ) expresion )? ';'
           | expresion ';'
           | 'return' expresion ';'
           | 'getline' '(' 'cin' ',' 'IDENT' ')' ';'
           | 'cout' '<<' ( expresion | 'endl' ) ( '<<' ( expresion | 'endl' ) )* ';'
           | proposicion-compuesta
           | 'if' '(' expresion ')' proposicion ( 'else' proposicion )?
           | 'while' '(' expresion ')' proposicion
```

referenced by:

- proposicion
- proposicion-compuesta

expresion:

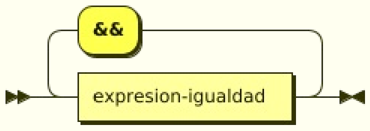


```
expresion
         ::= expresion-and ( '||' expresion-and )*
```

referenced by:

- expresion-atomica
- proposicion

expresion-and:

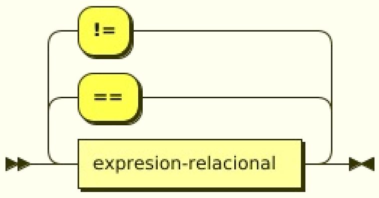


```
expresion-and
         ::= expresion-igualdad ( '&&' expresion-igualdad )*
```

referenced by:

- expresion

expresion-igualdad:

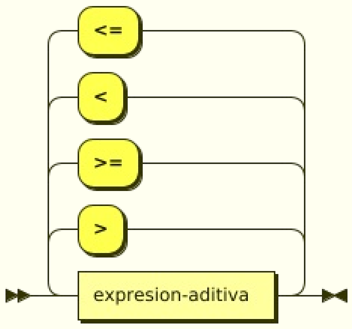


```
expresion-igualdad
         ::= expresion-relacional ( ( '==' | '!=' ) expresion-relacional )*
```

referenced by:

- expresion-and

expresion-relacional:

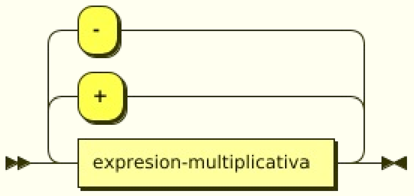


```
expresion-relacional
         ::= expresion-aditiva ( ( '>' | '>=' | '<' | '<=' ) expresion-aditiva )*
```

referenced by:

- expresion-igualdad

expresion-aditiva:

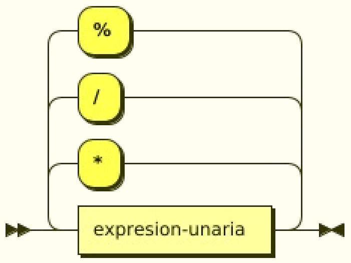


```
expresion-aditiva
         ::= expresion-multiplicativa ( ( '+' | '-' ) expresion-multiplicativa )*
```

referenced by:

- expresion-relacional

expresion-multiplicativa:

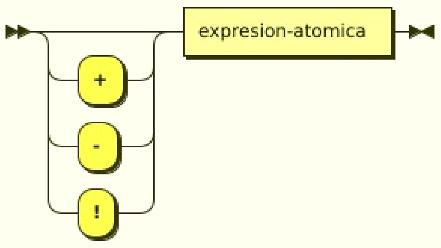


```
expresion-multiplicativa
         ::= expresion-unaria ( ( '*' | '/' | '%' ) expresion-unaria )*
```

referenced by:

- expresion-aditiva

expresion-unaria:

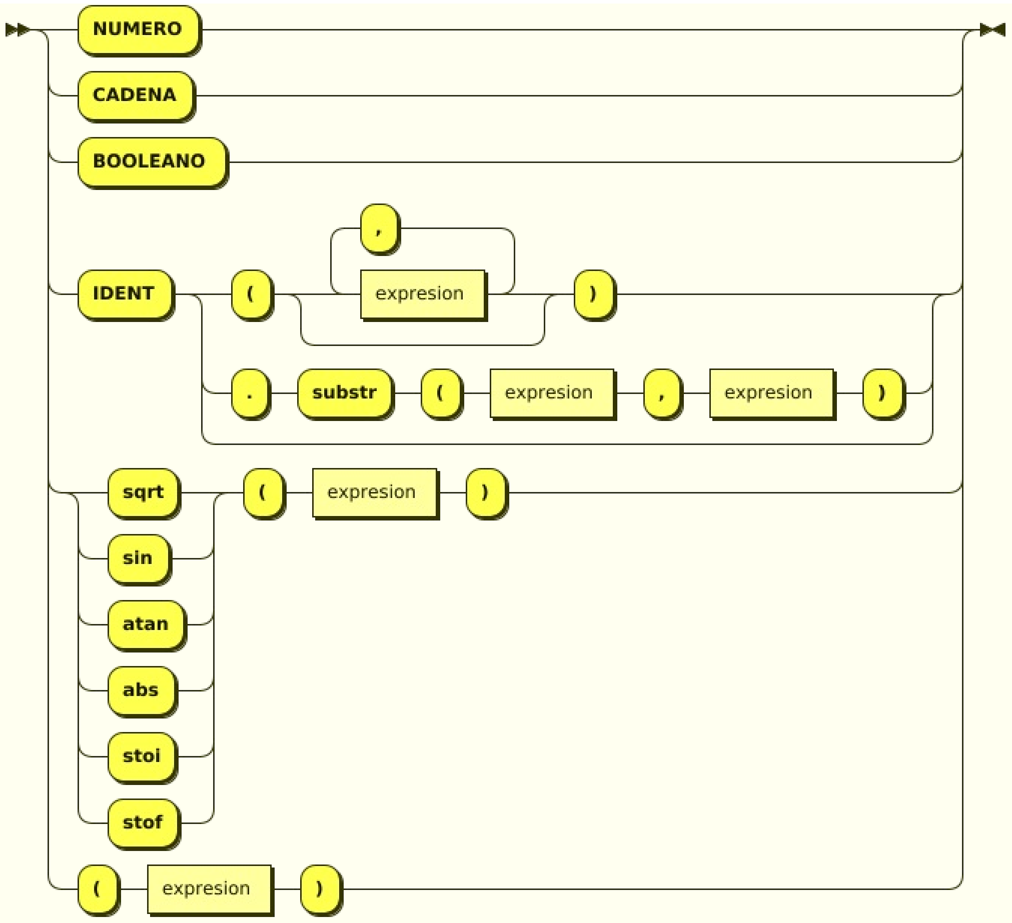


```
expresion-unaria
         ::= ( '+' | '-' | '!' )? expresion-atomica
```

referenced by:

- expresion-multiplicativa

expresion-atomica:



```
expresion-atomica
         ::= 'NUMERO'
           | 'CADENA'
           | 'BOOLEANO'
           | 'IDENT' ( '(' ( expresion ( ',' expresion )* )? ')' | '.' 'substr' '(' expresion ',' expresion ')' )?
           | ( 'sqrt' | 'sin' | 'atan' | 'abs' | 'stoi' | 'stof' ) '(' expresion ')'
           | '(' expresion ')'
```

referenced by:

- expresion-unaria