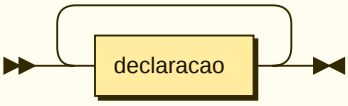
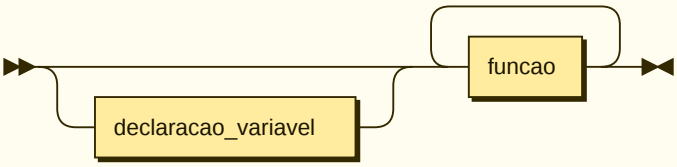
**programa:**




```
programa ::= declaracao+
```

no references

**declaracao:**

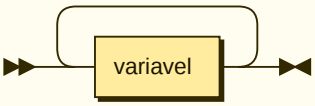



```
declaracao
        ::= declaracao_variavel? funcao+
```

referenced by:

- programa

**declaracao_variavel:**




```
declaracao_variavel
        ::= variavel+
```

referenced by:

- declaracao
- declaracao_composta

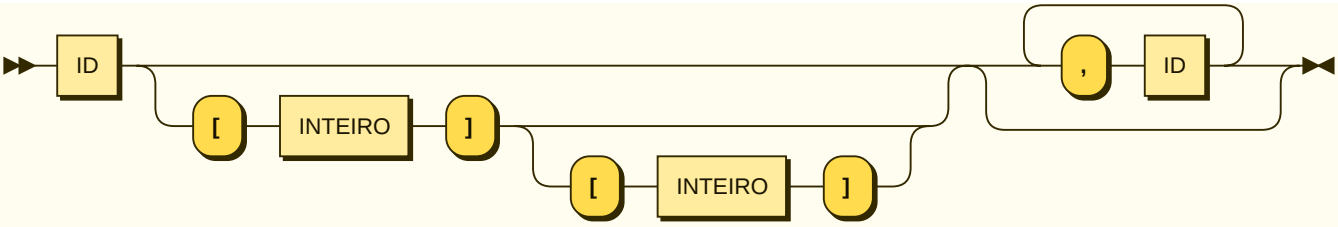
**variavel:**


```
variavel ::= tipo DOISPONTOS var
```

referenced by:

- declaracao_variavel

**var:**

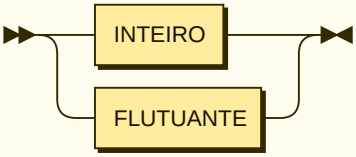



```
var      ::= ID ( '[' INTEIRO ']' ( '[' INTEIRO ']' )? )? ( ',' ID )*
```

referenced by:

- declaracao_atribuicao
- variavel

### tipo:

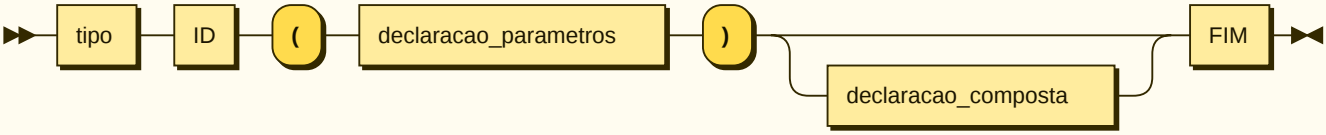



```
tipo     ::= INTEIRO
           | FLUTUANTE
```

referenced by:

- funcao
- parametro
- variavel

### funcao:

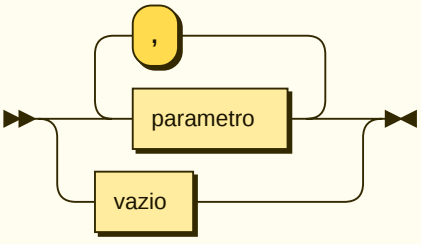



```
funcao   ::= tipo ID '(' declaracao_parametros ')' declaracao_composta? FIM
```

referenced by:

- declaracao

### declaracao_parametros:

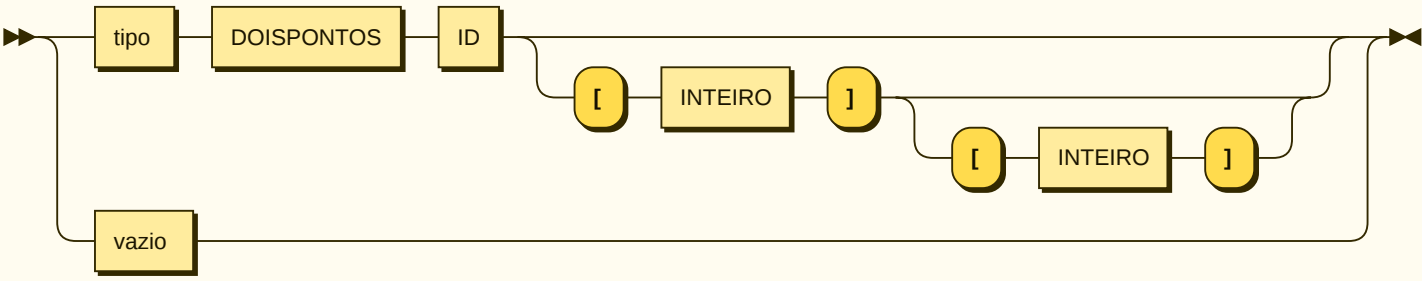



```
declaracao_parametros
         ::= parametro ( ',' parametro )*
           | vazio
```

referenced by:

- funcao

### parametro:

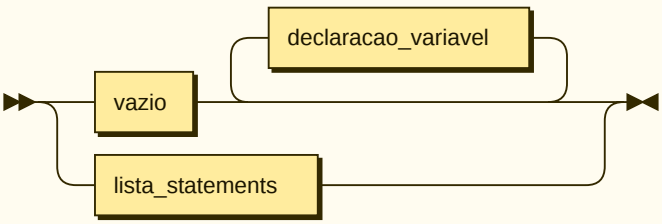



```
parametro
         ::= tipo DOISPONTOS ID ( '[' INTEIRO ']' ( '[' INTEIRO ']' )? )?
           | vazio
```

referenced by:

- declaracao_parametros

### declaracao_composta:

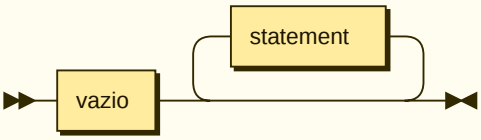



```
declaracao_composta
         ::= vazio declaracao_variavel*
           | lista_statements
```

referenced by:

- funcao
- statement

### lista_statements:

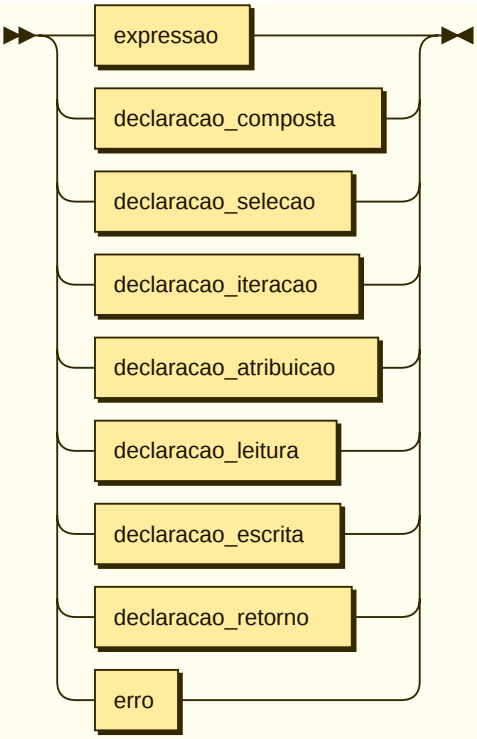



```
lista_statements
         ::= vazio statement*
```

referenced by:

- declaracao_composta
- declaracao_iteracao
- declaracao_selecao

### statement:

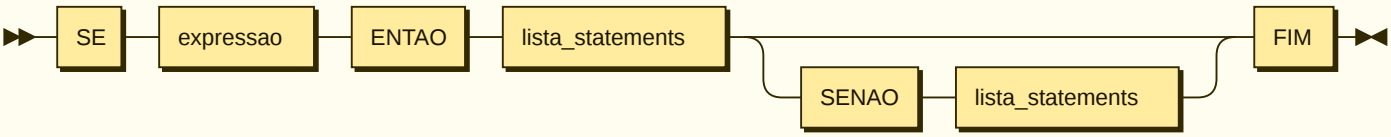



```
statement
         ::= expressao
           | declaracao_composta
           | declaracao_selecao
           | declaracao_iteracao
           | declaracao_atribuicao
           | declaracao_leitura
           | declaracao_escrita
           | declaracao_retorno
           | erro
```

referenced by:

- lista_statements

### declaracao_selecao:




```
declaracao_selecao
         ::= SE expressao ENTAO lista_statements ( SENAO lista_statements )? FIM
```

referenced by:

- statement

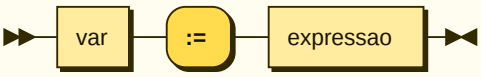
### declaracao_iteracao:


```
declaracao_iteracao
         ::= REPITA lista_statements ATE expressao
```

referenced by:

- statement

### declaracao_atribuicao:

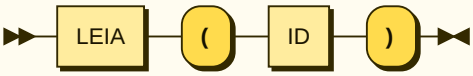



```
declaracao_atribuicao
         ::= var ':=' expressao
```

referenced by:

- statement

### declaracao_leitura:

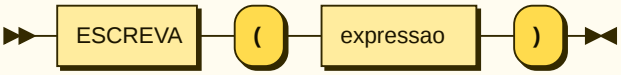



```
declaracao_leitura
         ::= LEIA '(' ID ')'
```

referenced by:

- statement

### declaracao_escrita:

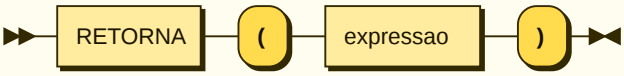



```
declaracao_escrita
         ::= ESCREVA '(' expressao ')'
```

referenced by:

- statement

### declaracao_retorno:

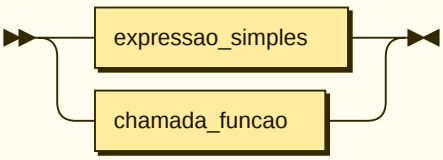



```
declaracao_retorno
         ::= RETORNA '(' expressao ')'
```

referenced by:

- statement

### expressao:

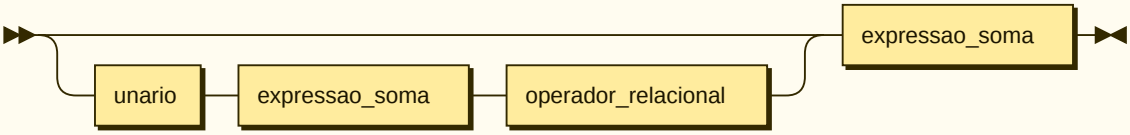


```
expressao
         ::= expressao_simples
           | chamada_funcao
```

referenced by:

- chamadafuncao
- declaracao_atribuicao
- declaracao_escrita
- declaracao_iteracao
- declaracao_retorno
- declaracao_selecao
- fator
- statement
- unario

### expressao_simples:

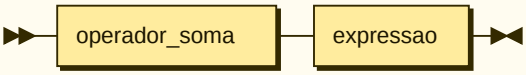



```
expressao_simples
         ::= ( unario expressao_soma operador_relacional )? expressao_soma
```

referenced by:

- expressao

### unario:

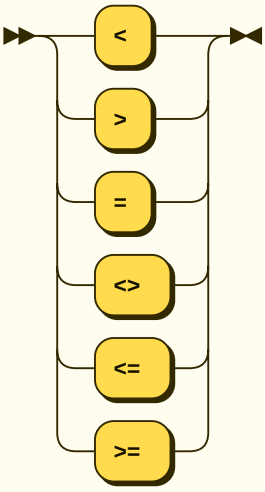



```
unario   ::= operador_soma expressao
```

referenced by:

- expressao_simples

### operador_relacional:



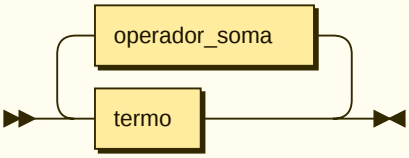

```
operador_relacional
         ::= '<'
           | '>'
           | '='
```

```
            | '<>'
            | '<='
            | '>='
```

referenced by:

- expressao_simples

### expressao_soma:

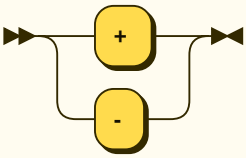



```
expressao_soma
         ::= termo ( operador_soma termo )*
```

referenced by:

- expressao_simples

### operador_soma:

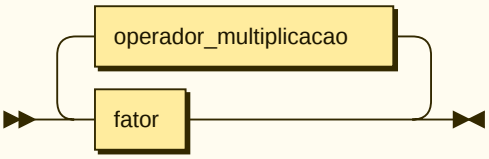



```
operador_soma
         ::= '+'
           | '-'
```

referenced by:

- expressao_soma
- unario

### termo:

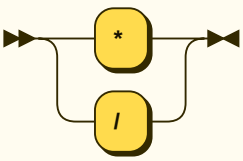



```
termo    ::= fator ( operador_multiplicacao fator )*
```

referenced by:

- expressao_soma

### operador_multiplicacao:

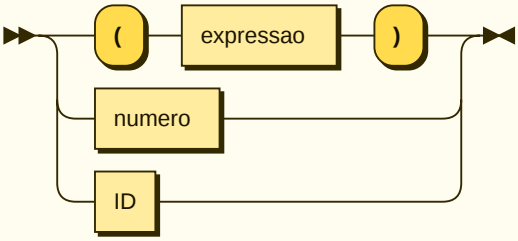


```
operador_multiplicacao
         ::= '*'
           | '/'
```

referenced by:

- termo

### fator:

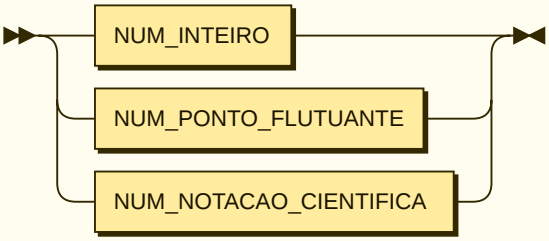



```
fator    ::= '(' expressao ')'
           | numero
           | ID
```

referenced by:

- termo

### numero:

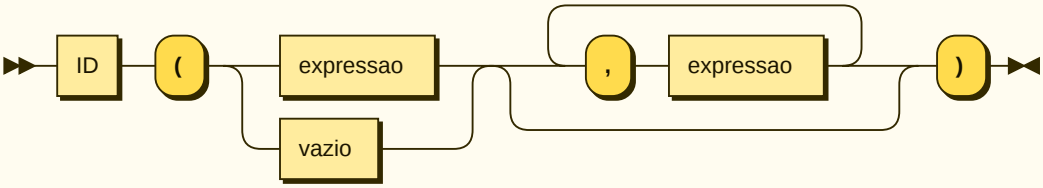



```
numero   ::= NUM_INTEIRO
           | NUM_PONTO_FLUTUANTE
           | NUM_NOTACAO_CIENTIFICA
```

referenced by:

- fator

### chamadafuncao:




```
chamadafuncao
         ::= ID '(' ( expressao | vazio ) ( ',' expressao )* ')'
```

no references

---

... generated by Railroad Diagram Generator