# Tokens Utilizados

## Palavras Reservadas

|  |  |
| --- | --- |
| void | PR\_VOID |
| int | PR\_INT |
| float | PR\_FLOAT |
| char | PR\_CHAR |
| bool | PR\_BOOL |
| if | PR\_IF |
| else | PR\_ELSE |
| for | PR\_FOR |
| while | PR\_WHILE |
| do | PR\_DO |
| return | PR\_RETURN |
| break | PR\_BREAK |
| continue | PR\_CONTINUE |
| goto | PR\_GOTO |
| true | PR\_TRUE |
| false | PR\_FALSE |

## Operadores

|  |  |
| --- | --- |
| + | OP\_ADICAO |
| - | OP\_SUBTRACAO |
| \* | OP\_MULTIPLICACAO |
| / | OP\_DIVISAO |
| % | OP\_MODULO |
| ? | OP\_TERNARIO |
| : | OP\_DOISPONTOS |
| ! | OP\_NEGACAO |
| & | OP\_ENDERECO |
| . | OP\_PONTO |
| -> | OP\_FLECHA |
| < | OP\_MENOR |
| > | OP\_MAIOR |
| == | OP\_IGUALDADE |
| != | OP\_DIFERENCA |
| <= | OP\_MENORIGUAL |
| >= | OP\_MAIORIGUAL |
| = | OP\_IGUAL |
| += | OP\_ADICAOIGUAL |
| -= | OP\_SUBTRACAOIGUAL |
| \*= | OP\_MULTIPLICACAOIGUAL |
| /= | OP\_DIVISAOIGUAL |
| %= | OP\_MODULOIGUAL |
| ++ | OP\_INCREMENTO |
| && | OP\_AND |
| || | OP\_OR |

## Sinais de Pontuação

|  |  |
| --- | --- |
| , | SP\_VIRGULA |
| ; | SP\_PONTOEVIRGULA |
| ( | SP\_ABREPARENTESES |
| ) | SP\_FECHAPARENTESES |
| [ | SP\_ABRECOLCHETES |
| ] | SP\_FECHACOLCHETES |
| { | SP\_ABRECHAVES |
| } | SP\_FECHACHAVES |

## Literais Básicos

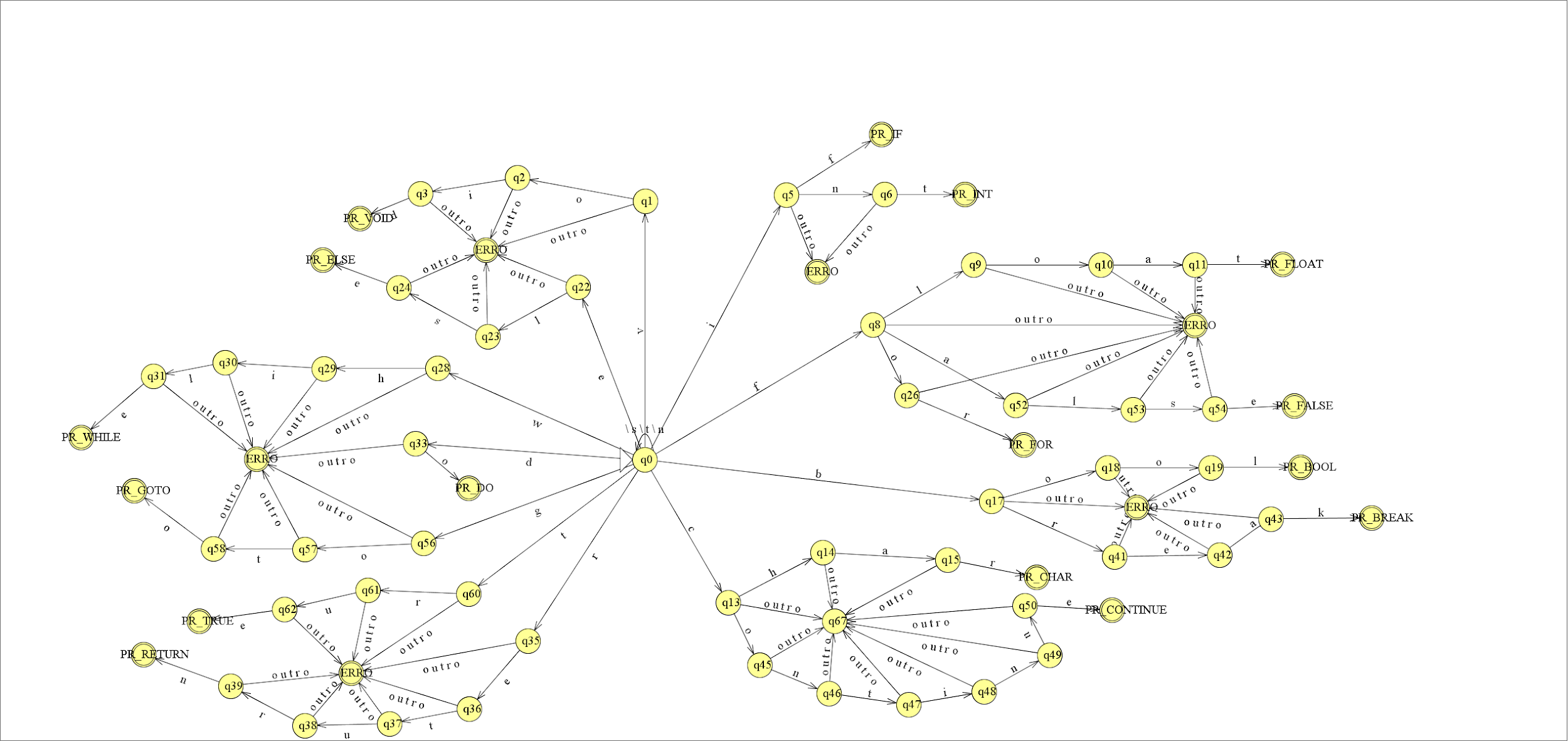
|  |  |
| --- | --- |
| inteiros | LB\_INT |
| reais | LB\_FLOAT |
| caracteres | LB\_CHAR |
| strings | LB\_STRING |
| booleanos | LB\_BOOL |

## Identificadores

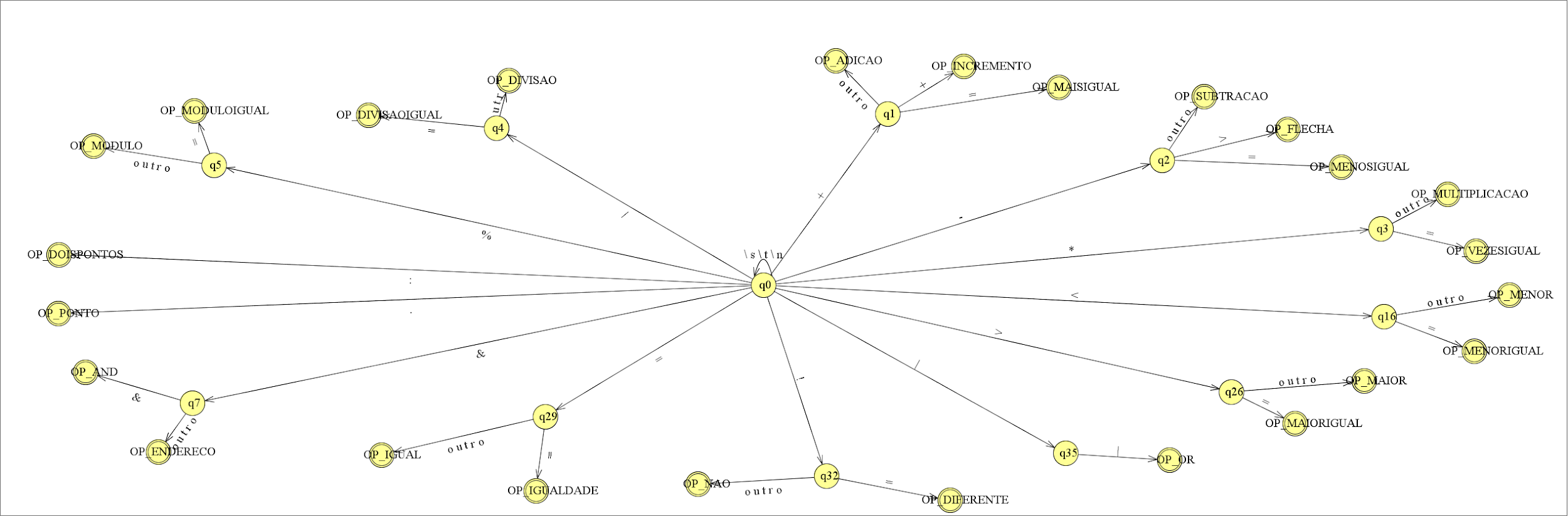
|  |  |
| --- | --- |
| identificadores | ID |

# Diagramas de Transição

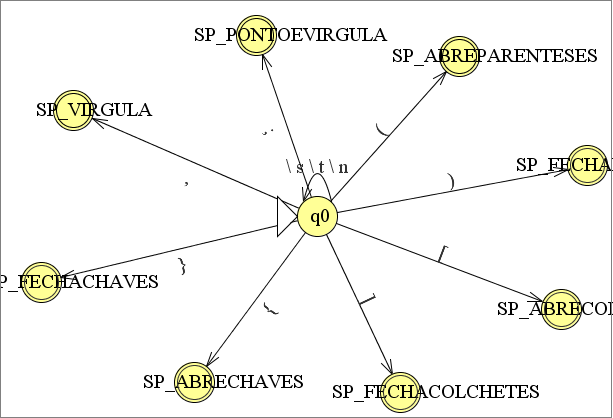
## Palavras Reservadas



## Operadores

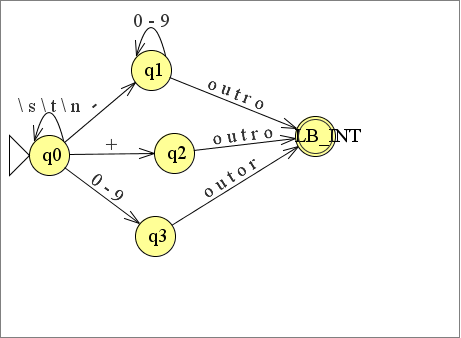


## Sinais de Pontuação

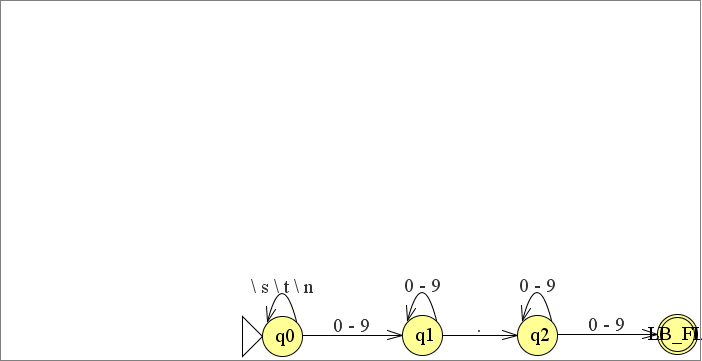


## Literais Básicos

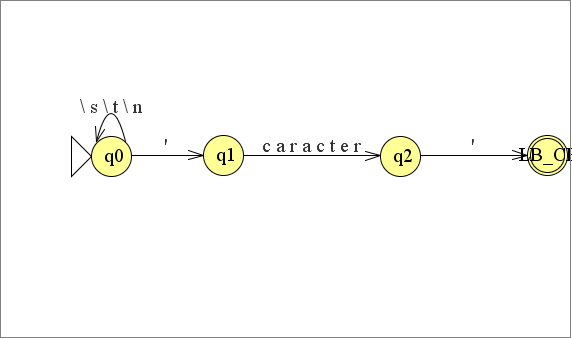
### Inteiros



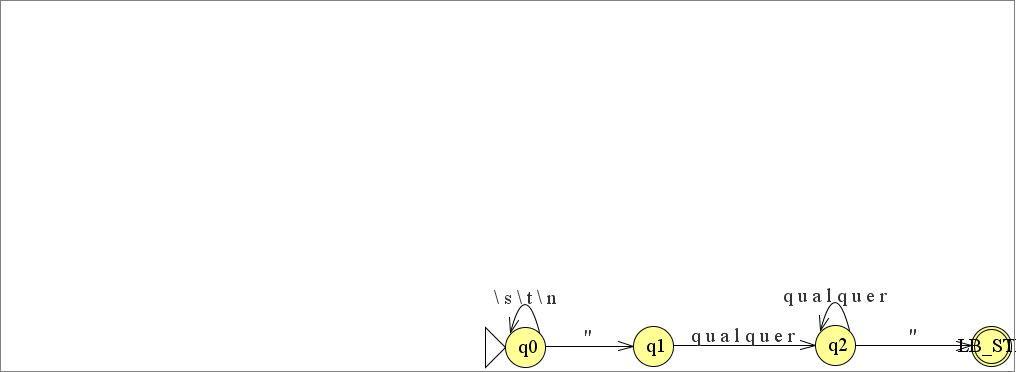
### Reais



### Caracteres



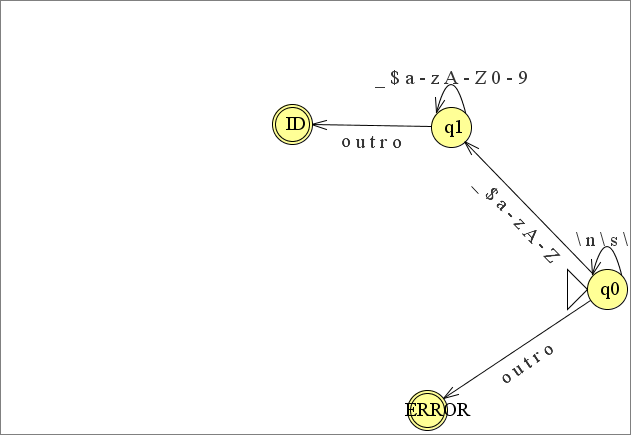
### Strings



### Booleanos



## Identificadores



# Técnicas utilizadas na gramática

programa → lista-decl

lista-decl → decl lista-decl | decl

decl → decl-var | decl-main

decl-var → VAR espec-tipo var ;

decl-main → MAIN ( ) bloco END

espec-tipo → INT | REAL | CHAR

bloco → lista-com

lista-com → comando lista-com | ɛ

comando → decl-var | com-atrib | com-selecao | com-repeticao | com-leitura | com-escrita

com-atrib → var = exp ;

com-leitura → SCAN ( var ) ; | SCANLN ( var ) ;

com-escrita → PRINT ( exp ) ; | PRINTLN ( exp ) ;

com-selecao → IF exp THEN bloco END-IF | IF exp THEN bloco ELSE bloco END-IF

com-repeticao → WHILE exp DO bloco LOOP

exp → exp-soma op-relac exp-soma | exp-soma

op-relac → <= | < | > | >= | == | <>

exp-soma → exp-mult op-soma exp-soma | exp-multi

op-soma → + | -

exp-mult → exp-simples op-mult exp-mult | exp-simples

op-mult → \* | / | DIV | MOD

exp-simples → ( exp ) | var | literal

literal → NUMINT | NUMREAL | CARACTERE | STRING

var → ID

|  |  |  |  |
| --- | --- | --- | --- |
| **APS** | **P** | **P+** | **t** |
| programa | lista-decl | Prim(lista-decl) | VAR, MAIN |
| lista-decl | decl | Prim(decl) | VAR, MAIN |
| decl | decl-var, decl-main | Prim(decl-var), Prim(decl-main) | VAR, MAIN |
| decl-var | VAR | VAR | VAR |
| decl-main | MAIN | MAIN | MAIN |
| espec-tipo | INT, REAL, CHAR | INT, REAL, CHAR | INT, REAL, CHAR |
| bloco | lista-com | Prim(lista-com) | VAR, ID, IF, IF, WHILE, SCAN, SCANL, PRINT, PRINTLN, END |
| lista-com | comando, ɛ | Prim(comando), Segue(lista-com) -> Segue(bloco) -> END | VAR, ID, IF, IF, WHILE, SCAN, SCANL, PRINT, PRINTLN, END |
| comando | decl-var, com-atrib, com-selecao, com-repeticao, com-leitura, com-escrita | Prim(decl-var), Prim(com-atrib), Prim(com-selecao), Prim(com-repeticao), Prim(com-leitura), Prim(com-escrita), | VAR, ID, IF, IF, WHILE, SCAN, SCANL, PRINT, PRINTLN |
| com-atrib | var | Prim(var) | ID |
| com-leitura | SCAN, SCANLN | SCAN, SCANLN | SCAN, SCANLN |
| com-escrita | PRINT, PRINTLN | PRINT, PRINTLN | PRINT, PRINTLN |
| com-selecao | IF, IF | IF, IF | IF, IF |
| com-repeticao | WHILE | WHILE | WHILE |
| exp | exp-soma | Prim(exp-soma) | (, NUMINT, NUMREAL, CARACTER, STRING, ID |
| op-relac | <=, <, >, >=, ==, <> | <=, <, >, >=, ==, <> | <=, <, >, >=, ==, <> |
| exp-soma | exp-mult | Prim(exp-mult) | (, NUMINT, NUMREAL, CARACTER, STRING, ID |
| op-soma | +, - | +, - | +, - |
| exp-mult | exp-simples | Prim(exp-simples) | (, NUMINT, NUMREAL, CARACTER, STRING, ID |
| op-mult | \*, /, DIV, MOD | \*, /, DIV, MOD | \*, /, DIV, MOD |
| exp-simples | (, var, literal | (, Prim(var), Prim(literal) | (, NUMINT, NUMREAL, CARACTER, STRING, ID |
| literal | NUMINT, NUMREAL, CARACTER, STRING | NUMINT, NUMREAL, CARACTER, STRING | NUMINT, NUMREAL, CARACTER, STRING |
| var | ID | ID | ID |

# Mensagens de Erro