# 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

# 

# Técnicas utilizadas na gramática

1. programa → lista-decl
2. lista-decl → decl lista-decl | decl
3. decl → decl-var | decl-main
4. decl-var → VAR espec-tipo var ;
5. decl-main → MAIN ( ) bloco END
6. espec-tipo → INT | REAL | CHAR
7. bloco → lista-com
8. lista-com → comando lista-com | ɛ
9. comando → decl-var | com-atrib | com-selecao | com-repeticao | com-leitura | com-escrita
10. com-atrib → var = exp ;
11. com-leitura → SCAN ( var ) ; | SCANLN ( var ) ;
12. com-escrita → PRINT ( exp ) ; | PRINTLN ( exp ) ;
13. com-selecao → IF exp THEN bloco END-IF | IF exp THEN bloco ELSE bloco END-IF
14. com-repeticao → WHILE exp DO bloco LOOP
15. exp → exp-soma op-relac exp-soma | exp-soma
16. op-relac → <= | < | > | >= | == | <>
17. exp-soma → exp-mult op-soma exp-soma | exp-mult
18. op-soma → + | -
19. exp-mult → exp-simples op-mult exp-mult | exp-simples
20. op-mult → \* | / | DIV | MOD
21. exp-simples → ( exp ) | var | literal
22. literal → NUMINT | NUMREAL | CARACTERE | STRING
23. var → ID

|  |  |  |  |
| --- | --- | --- | --- |
| **APS** | **P** | **P+** | **t** |
| programa | Prim(lista-decl) |  |  |
| lista-decl | Prim( |  |  |
| decl |  |  |  |
| decl-var |  |  |  |
| decl-main |  |  |  |
| espec-tipo |  |  |  |
| bloco |  |  |  |
| lista-com |  |  |  |
| comando |  |  |  |
| com-atrib |  |  |  |
| com-leitura |  |  |  |
| com-escrita |  |  |  |
| com-selecao |  |  |  |
| com-repeticao |  |  |  |
| exp |  |  |  |
| op-relac |  |  |  |
| exp-soma |  |  |  |
| op-soma |  |  |  |
| exp-mult |  |  |  |
| op-mult |  |  |  |
| exp-simples |  |  |  |
| literal |  |  |  |
| var |  |  |  |

# Mensagens de Erro