



```

<comando> ::=
  read ( <var_read> ) |
  write ( <var_write> ) |
  for <id> {A25} := <expressao> {A26} to {A27} <expressao> {A28}
  do begin <sentencas> end {A29} |
  repeat {A23} <sentencas> until ( <condicao> ) {A24} |
  while {A20} ( <condicao> ) {A21} do begin <sentencas> end {A22} |
  if ( <condicao> ) {A17} then begin <sentencas> end {A18}
  <pfalsa> {A19} |
  <id> {A13} := <expressao> {A14} |

```

---

*<chamada\_procedimento>*

---

*<chamada\_procedimento> ::= <id\_proc> <argumentos>*

*<argumentos> ::= ( <lista\_arg> ) |  $\lambda$*

*<lista\_arg> ::= <expressao> <cont\_lista\_arg>*

*<cont\_lista\_arg> ::= , <lista\_arg> |  $\lambda$*

---

*<condicao> ::= <expressao> <relacao> {A15} <expressao> {A16}*

*<pfalsa> ::= else begin <sentencas> end |  $\lambda$*

*<relacao> ::= = | > | < | >= | <= | <>*

*<expressao> ::= <termo> <outros\_termos>*

*<outros\_termos> ::= <op\_ad> {A9} <termo> {A10} <outros\_termos> |  $\lambda$*

*<op\_ad> ::= + | -*

*<termo> ::= <fator> <mais\_fatores>*

*<mais\_fatores> ::= <op\_mul> {A11} <fator> {A12} <mais\_fatores> |  $\lambda$*

*<op\_mul> ::= \* | /*

---

*<fator> ::= <id\_var> {A7} | <intnum> {A8} |*

*( <expressao> ) | <id\_funcao> <argumentos>*

---

*<id> ::= letra (letra | digito)\* {A3}*

*<intnum> ::= digito+ {A4}*

## Observação sobre a sintaxe da Linguagem Pascal

A construção de programas poderá conter comentários, os quais são inicializados pelo símbolo "{" e finalizados pelo símbolo "}". Qualquer seqüência de caracteres contidos entre estes dois delimitadores deverão ser desconsiderados pelo compilador. Portanto, durante a fase de análise léxica, torna-se viável desprezar todos os comentários. Isto pode ser feito da seguinte forma:

```
Se NextCar = '{' então
    Enquanto NextCar <> '}' faça
        Chamar Ler
    Fim faça
    Chamar Ler
Fim se
```

## Exemplo de um programa em PASCAL SIMPLIFICADO

```
{ Programa: Somatório }  
{ Autor   : Rogério Melo Nepomuceno }  
{ Data    : 21 de Setembro de 1999 }
```

**Program Somatorio**

**Var**

```
Ini, Fim, Conta, Soma : integer;
```

**Begin**

```
Ini := 0;
```

```
Fim := 0;
```

```
Soma := 0;
```

```
read( Ini, Fim );
```

```
For Conta := Ini To Fim do begin
```

```
    Soma := Soma + Conta
```

```
End;
```

```
write( Soma );
```

**End.**