**<bool-lit> ::= true | false**

<comando> ::= <condicional> | <iterativo> | <comando-composto> | <id>((“[“<expressão>”]”)\* **:=** <expressão> | ("(" (<lista-de-expressões> | <vazio>) ")") OBS

<comando-composto> ::= begin (<comando>;)\* end

<condicional> ::= if <expressão> then <comando> ( else <comando> | <vazio> )

<corpo> ::= (<declaração>;)\* <comando-composto>

<declaração> ::= <declaração-de-variável> | <declaração-de-função> | <declaração-de-procedimento>

<declaração-de-função> ::= function <id> "(" <lista-de-parâmetros> | <vazio> ")" : <tipo-simples> ; <corpo>

<declaração-de-procedimento> ::= procedure <id> "(" <lista-de-parâmetros> | <vazio> ")" ; <corpo>

<declaração-de-variável> ::= var <lista-de-ids> : <tipo>

**<digito> ::= 0 | 1 | 2 | ...| 9**

<expressão> ::= <expressão-simples>(<op-erl><expressão-simples>|vazio)

<expressão-simples> ::= <termo>(<op-ad><termo>)\*

<fator> ::= (<bool-lit> | <int-lit> | <float-lit>) | "(" <expressão> ")" | <id>((“[“<expressão>”]”)\* | ("(" (<lista-de-expressões> | <vazio>) ")")

OBS: (SE LIGAR NA HORA DO ID SEPARAR – VARIAVEL E CHAMADA DE FUNÇÂO)

**<float-lit> ::= Digit(Digit)\*. Digit(Digit)\*| Digit(Digit)\*. | . Digit(Digit)\***

**<id> ::= Letter(Letter|Digit)\***

**<int-lit> ::= Digit(Digit)\***

<iterativo> ::= while <expressão> do <comando>

**<letra> ::= a | b | c | ... | z**

<lista-de-expressões> ::= <expressão>(,<expressão>)\*

<lista-de-ids> ::= <id> (,<id>)\*

<lista-de-parâmetros> ::= <parâmetros>(;<parâmetros>)\*

**<op-ad> ::= + | - | or**

**<op-mul> ::= \* | / | and**

**<op-rel> ::= < | > | <= | >= | = | <>**

<parâmetros> ::= ( var | <vazio> ) <lista-de-ids> : <tipo-simples>

<programa> ::= **program** <id> **;** <corpo>

<termo> ::= <fator>(<op-mul><fator>)\*

<tipo> ::= <tipo-agregado> | <tipo-simples>

<tipo-agregado> ::= **array [** **int-lit** .. **int-lit** **] of** <tipo> **OBS:**  **[** **int-li**>=**intlit** **]**

<tipo-simples> ::= **integer** | **real** | **boolean**

**<vazio> ::= ε**