

```

<atribuição> ::=
    <variável> := <expressão>
<bool-lit> ::=
    true
    | false
<chamada-de-função> ::=
    <id> "(" <lista-de-expressões> | <vazio> ")"
<chamada-de-procedimento> ::=
    <id> "(" <lista-de-expressões> | <vazio> ")"
<char>
    ::= <letra>
    | <digito>
    | <outros>
<char-lit> ::=
    '<char>'
<comando> ::=
    <atribuição>
    | <condicional>
    | <iterativo>
    | <chamada-de-procedimento>
    | <comando-composto>
<comando-composto> ::=
    begin <lista-de-comandos> end
<condicional> ::=
    if <expressão> then <comando> ( else <comando> | <vazio> )
<corpo> ::=
    <declarações> <comando-composto>
<declaração> ::=
    <declaração-de-variável>
    | <declaração-de-constante>
    | <declaração-de-função>
    | <declaração-de-procedimento>
<declaração-de-constante> ::=
    const <lista-de-ids> = <literal>
<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>
<declarações> ::=
    <declaração> ;
    | <declarações> <declaração> ;
    | <vazio>
<digito> ::=
    0
    | 1
    | 2
    | ...
    | 9
<expressão> ::=
    <expressão-simples>
    | <expressão-simples> <op-rel> <expressão-simples>
<expressão-simples> ::=
    <expressão-simples> <op-ad> <termo>
    | <termo>
<fator> ::=
    <variável>
    | <literal>
    | "(" <expressão> ")"
    | <chamada-de-função>
<float-lit> ::=
    <int-lit> . <int-lit>

```

```

    | <int-lit> .
    | . <int-lit>
<id> ::=
    <letra>
    | <id> <letra>
    | <id> <digito>
<int-lit> ::=
    <digito>
    | <int-lit> <digito>
<iterativo> ::=
    while <expressão> do <comando>
<letra> ::=
    a
    | b
    | c
    | ...
    | z
<lista-de-comandos> ::=
    <comando> ;
    | <lista-de-comandos> <comando> ;
    | <vazio>
<lista-de-expressões> ::=
    <lista-de-expressões> , <expressão>
    | <expressão>
<lista-de-ids> ::=
    <id>
    | <lista-de-ids> , <id>
<lista-de-parâmetros> ::=
    <parâmetros>
    | <lista-de-parâmetros> ; <parâmetros>
<literal> ::=
    <bool-lit>
    | <int-lit>
    | <float-lit>
    | <char-lit>
<op-ad> ::=
    +
    | -
    | or
<op-mul> ::=
    *
    | /
    | and
<op-rel> ::=
    <
    | >
    | <=
    | >=
    | =
    | <>
<outros> ::=
    !
    | @
    | #
    | ...
<parâmetros> ::=
    ( var | <vazio> ) <lista-de-ids> : <tipo-simples>
<programa> ::=
    program <id> ; <corpo> .
<seletor> ::=
    <seletor> "[" <expressão> "]"
    | "[" <expressão> "]"
    | <vazio>

```

```
<termo> ::=
    <termo> <op-mul> <fator>
    | <fator>
<tipo> ::=
    <tipo-agregado>
    | <tipo-simples>
<tipo-agregado> ::=
    array [ <literal> .. <literal> ] of <tipo>
<tipo-simples> ::=
    integer
    | real
    | char
    | boolean
<variável> ::=
    <id> <seletor>
<vazio> ::=
    ε
```