

```

<programa> ::= <declara> <rotina> <sentencas> END
<declara> ::= <dvar> <declara> | λ
<dvar> ::= dim <variaveis> as <tipo_var>
<tipo_var> ::= integer
<variaveis> ::= <id> <mais_var>
<mais_var> ::= , <variaveis> | λ
<rotina> ::= <procedimento> | <funcao> | λ
<procedimento> ::= sub <id> <parametros> <sentencas> end sub
    <rotina>
<funcao> ::= function <id> <parametros> as <tipo_funcao>
    <sentencas> end function <rotina>
<parametros> ::= ( <lista_parametros> ) | λ
<lista_parametros> ::= <id> as <tipo_var> <cont_lista_par> | λ
<cont_lista_par> ::= , <id> as <tipo_var> <cont_lista_par> | λ
<tipo_funcao> ::= integer
<sentencas> ::= <comando> <mais_sentencas>
<mais_sentencas> ::= <sentencas> | λ
<var_read> ::= <id> <mais_var_read>
<mais_var_read> ::= , <var_read> | λ
<var_write> ::= <id> <mais_var_write>
<mais_var_write> ::= , <var_write> | λ
<comando> ::= input <var_read> | print <var_write> | for <id> =
    <expressao> to <expressao> <sentencas> next | do loop
    <sentencas> while <condicao> | do while <condicao>
    <sentencas> loop | if <condicao> then <sentencas> <pfalsa>
    end if | let <idvar> = <expressao> | <chamada_procedimento>
<chamada_procedimento> ::= call <id_proc> <argumentos>
<argumentos> ::= ( <lista_arg> ) | λ
<lista_arg> ::= <expressao> <cont_lista_arg>
<cont_lista_arg> ::= , <lista_arg> | λ
<pfalsa> ::= else <sentencas> | λ
<condicao> ::= <expressao> <relacao> <expressao>
<relacao> ::= = | <> | >= | <= | > | <
<expressao> ::= <termo> <outros_termos>
<outros_termos> ::= <op_ad> <termo> <outros_termos> | λ
<op_ad> ::= + | -
<termo> ::= <fator> <mais_fatores>
<mais_fatores> ::= <op_mul> <fator> <mais_fatores> | λ
<op_mul> ::= * | /
<fator> ::= <id_var> | <intnum> | (<expressao>) | <id_funcao>
    <argumentos>
<id> ::= letra (letra | digito)*
<intnum> ::= digito+

```