

```

programa
    : declaracao_programa (declaracao_constantes)? ( declaracao_variaveis )? (
declaracao_rotinas )? bloco

declaracao_programa
    : "programa" IDENTIFICADOR ";"

declaracao_variaveis
    : "variaveis" ( IDENTIFICADOR ( "," IDENTIFICADOR )* ":" tipo ( "<-" literal
)? ";" )+

declaracao_constantes
    : "constantes" ( IDENTIFICADOR ( ":" tipo )? "<-" literal ";" )+

assinatura_rotina
    : IDENTIFICADOR ( "(" ( ( "ref" | "const" )? IDENTIFICADOR ( ","
IDENTIFICADOR)* ":" parametro ( ";" ( "ref" | "const" )? IDENTIFICADOR ( ","
IDENTIFICADOR)* ":" parametro )* )? ")" )?

declaracao_funcao
    : "funcao" assinatura_rotina ":" tipo ";" bloco

declaracao_procedimento
    : "procedimento" assinatura_rotina ";" bloco

declaracao_rotinas
    : "rotinas" ( declaracao_funcao | declaracao_procedimento )+

bloco
    : ( declaracao_constantes )? ( declaracao_variaveis )? "inicio" ( comando )*
"fim" ";"

comandos
    : ( "inicio" ( comando )* "fim" ";" ) | comando

tipo_primitivo
    : "inteiro" | "real" | "caractere" | "texto" | "logico"

tipo_arranjo
    : "arranjo" "[" ( INTEIRO_LITERAL ) ( "," INTEIRO_LITERAL )* "]" "de"
tipo_primitivo

tipo
    : tipo_primitivo | tipo_arranjo

parametro_tipo_arranjo
    : "arranjo" "[" ( "," )* "]" "de" tipo_primitivo

parametro
    : tipo_primitivo | parametro_tipo_arranjo

posicao_memoria
    : IDENTIFICADOR ( "[" ( expressao ( "," expressao )* ) "]" )?

```

```

invocacao_rotina
    : IDENTIFICADOR ( "(" ( expressao ( "," expressao ) )? ")" )?

literal_primitivo
    : LITERAL_INTEIRO | LITERAL_REAL | LITERAL_CARACTER | LITERAL_TEXTO |
    LITERAL_LOGICO

lista_literais_primitivos
    : literal_primitivo ( "," literal_primitivo )*

lista_literal_arranjo
    : literal_arranjo ( "," literal_arranjo )*

literal_arranjo
    : "(" ( lista_literais_primitivos | lista_literal_arranjo ) ")"

literal
    : literal_primitivo | literal_arranjo

expressao_prioritaria
    : "(" expressao ")"

expressao_unaria
    : ( "neg" | "inc" | "dec" | "-" | "+" ) expressao

expressao_binaria
    : expressao ( "ou" | "ouex" | "e" | "=" | "#" | ">" | "<" | ">=" | "<=" | "*"
    | "+" | "/" | "-" | "mod" ) expressao

expressao
    : posicao_memoria | invocacao_rotina | literal_primitivo | expressao_unaria
    | expressao_binaria | expressao_prioritaria

comando_invocar_rotina
    : invocacao_rotina ";"

comando_encerrar
    : "encerrar" ";"

comando_retonar
    : "retorne" expressao ";"

comando_atribuicao
    : posicao_memoria "<-" expressao

comando_parar
    : "parar" ";"

comando_continuar
    : "continuar" ";"

comando_negacao
    : "neg" expressao ";"

```

comando_incremento
:"inc" expressao ";"

comando_decremento
:"dec" expressao ";"

comando_menos
:"- " expressao ";"

comando_mais
:"+ " expressao ";"

comando
: comando_invocar_rotina | comando_encerrar | comando_retonar |
comando_atribuicao | comando_parar | comando_continuar | comando_negacao |
comando_incremento | comando_decremento | comando_menos | comando_mais

estrutura_se
:"se" expressao "entao" comandos ("senao" comandos)?

estrutura_caso
:"caso" expressao "seja" "inicio" (literal_primitivo ":" comandos)+ ("senao" comandos)? "fim" ";"

estrutura_enquanto
:"enquanto" expressao "faca" comandos

estrutura_repita
:"repita" comandos "ate" expressao ";"