

Início

```
Início = ListaFunc Início  
      | ListaProc Início  
      | ListaVar Início  
      | ListaFunc  
      | ListaProc  
      | ListaVar
```

```
ListaFunc = Função ListaFunc  
          | Função
```

```
ListaProc = Proc ListaProc  
          | Proc
```

```
ListaVar = Variável ListaVar  
         | Variável
```

Função

```
Função = 'func' Tipo 'id' '(' Parâmetros ')' FunçãoF  
FunçãoF = Bloco | ';' 
```

Procedimento

```
Proc = 'proc' Tipo 'id' '(' Parâmetros ')' ProcF  
ProcF = Bloco | ';' 
```

Variável

```
Variável = Tipo ListaId ';'   
         | 'const' Tipo ListaAtr ';' 
```

```
ListaAtr = 'id' ListaAtrF ListaAtrR  
ListaAtrR = ',' 'id' ListaAtrF ListaAtrR  
          | ε
```

```
ListaAtrF = Atribuição  
          | Array Atribuição
```

Identificadores

```
ListaId = Id ListaIdR  
ListaIdR = ',' Id ListaIdR  
         | ε
```

```
Id = 'id' IdF  
IdF = '(' ListaArgs ')' | Atribuição | Array IdFF | ε
```

$IdFF = Atribuição \mid \varepsilon$

$Array = '[' \ ArrayF$

$ArrayF = ']' \mid ExprBool \ ']'$

Bloco

$Bloco = '{' \ ListaSenten\c{c}as \ '}'$

Senten\c{c}as

$ListaSenten\c{c}as = Senten\c{c}a \ ListaSenten\c{c}as$
 $\mid Senten\c{c}a$

$Senten\c{c}a = If \mid While \mid For \mid Desvio \mid Return \mid Entrada \mid Saída$
 $\mid Atribui\c{c}ão \mid ChamadaProc \mid \varepsilon$

$Entrada = 'input' \ ListaArgs \ ';'$

$Saída = 'print' \ CteStr \ ListaArgs \ ';'$

$If = 'if' \ '(' \ ExprBool \ ')' \ Bloco \ ElseIf \ Else$

$ElseIf = 'else \ if' \ '(' \ ExprBool \ ')' \ Bloco \ ElseIf$
 $\mid \varepsilon$

$Else = 'else' \ Bloco$
 $\mid \varepsilon$

$While = 'while' \ '(' \ ExprBool \ ')' \ Bloco$

$For = 'for' \ Id \ 'in' \ '(' \ ExprBool \ ',' \ ExprBool \ ')' \ 'step' \ ExprBool \ Bloco$

$Desvio = 'break' \ ';'$

$Return = 'return' \ ExprBool \ ';'$

$Atribui\c{c}ão = OpAtribui\c{c}ão \ ExprBool \ ';'$

$ChamadaProc = 'id' \ '(' \ ListaArgs \ ')' \ ';'$

Parâmetros

$Parâmetros = ListaParam$
 $\mid \varepsilon$

$ListaParam = Tipo \ Id \ ListaParamR$

$ListaParamR = ',' \ Tipo \ Id$

| ε

Argumentos

Argumentos = *ListaArgs* | ε

ListaArgs = *ExprBool* *ListaArgsR*

ListaArgsR = *' , '* *ExprBool*

| ε

Tipos

Tipo = *'int'* | *'float'* | *'char'* | *'bool'* | *'string'*

Expressões

ExprBool = *TermoBool* *ExprBoolR*

ExprBoolR = *OpLogic* *TermoBool*

| ε

TermoBool = *ExprConcat* *OpRelac* *ExprConcat*

| *ExprConcat*

| *'!'* *TermoBool*

ExprConcat = *ExprAritm* *ExprConcatR*

ExprConcatR = *OpConcat* *ExprAritm*

| ε

ExprAritm = *TermoAritm* *ExprAritmR*

ExprAritmR = *OpAritm* *TermoAritm*

| ε

TermoAritm = *FatorAritm* *TermoAritmR*

TermoAritmR = *OpMult* *FatorAritm*

| ε

FatorAritm = *'('* *FatorAritmF*

| *Id*

| *CteInt*

| *CteFloat*

| *CteChar*

| *CteBool*

| *CteStr*

| *'['* *ListaArray* *']'*

FatorAritmF = *Tipo* *')* *ExprBool*

| *ExprBool* *')*

```
ListaArray = ExprBool ListaArrayR
ListaArrayR = ',' ExprBool
            | ε
```

Operadores

```
OpUnario    = OpMenos | OpNot
OpRelac      = OpMaior | OpMenor | OpMaiorEq | OpMenorEq | OpEq | OpDifer
OpLogic      = OpAnd | OpOr
OpAdit       = OpMais | OpMenos
OpMult       = OpAster | OpDiv | OpMod
```

```
OpEq         = '=='
OpMaior      = '>'
OpMenor      = '<'
OpMaiorEq    = '>='
OpMenorEq    = '<='
OpDifer      = '!='
OpAnd        = 'and'
OpOr         = 'or'
OpNot        = '!'
OpMais       = '+'
OpMenos      = '-'
OpAster      = '*'
OpDiv        = '/'
OpMod        = '%'
OpAtr        = '='
OpConcat     = '++'
```