

## Lista de Tokens utilizados

<OP\_UN> = {!}  
<OP\_BIN> = { \* , / , && , || , < , > , <= , >= , == , != , % }  
<OP\_UN\_BIN> = { + , - }  
<CHARACTER> = ' ( a | b | ... | z | 0 | ... | 9 | ... ) '  
<VAR> = ( a | b | ... | z )<sup>+</sup>  
<NUM> = ( 0 | 1 | ... | 9 )<sup>\*</sup> ( . ) ? ( 0 | 1 | ... | 9 )<sup>+</sup>  
<AP> = { ( }  
<FP> = { ) }  
<AC> = { { }  
<FC> = { } }  
<PT\_VIRG> = { ; }  
<DOIS\_PT> = { : }  
<IGUAL> = { = }  
<WHILE> = while  
<FOR> = for  
<IF> = if  
<SWITCH> = switch  
<CASE> = case  
<DO> = do  
<EOF> = EOF

## Gramática

programa -> comando programa  
          | comando

comando -> while  
          | if  
          | swicth\_case  
          | do\_while  
          | for  
          | atribuicao <PT\_VIRG>

exp -> exp operador1 exp  
      | operador2 exp  
      | <AP> exp <FP>  
      | <VAR>  
      | <NUM>  
      | <SINAL><NUM>

operador1 -> <OP\_UN\_BIN>  
          | <OP\_BIN>

operador2 -> <OP\_UN\_BIN>  
          | <OP\_UN>

while -> <WHILE><AP> exp <FP> bloco

bloco -> <AC> programa <FC>  
| comando

if -> <IF><AP> exp <FP> bloco

switch\_case -> <SWITCH><AP><VAR><FP><AC> listaCase <FC>

case -> <CASE> identCase <DOIS\_PT> bloco listaCase

listaCase -> λ  
| case listaCase

identCase -> <NUM>  
| <VAR>  
| <CHARACTER>

atribuicao -> <VAR><IGUAL> exp

for -> <FOR><AP> atribuicao <PT\_VIRG> exp <PT\_VIRG> atribuicao <FP> bloco

do\_while -> <DO><AC> bloco <FC><WHILE><AP> exp <FP>