

programa -> listaComandos

listaComandos-> comando listaComandos | lambda

comando -> if | while | doWhile | for | switch | <PTV> | expressão<PTV> | lambda

while -> <WHILE> <AP> expressão <FP> comando | <WHILE> <AP> expressão <FP> <AC>  
listaComandos <FC>

doWhile -> <DO> <ACH> listaComandos <FCH> <WHILE> <AP> expressão <FP> <PTV> |  
<DO> comando <WHILE> <AP> expressão <FP> <PTV>

for -> <FOR> <AP> atribuição <PTV> expressão <PTV> atribuição <FP> comando |  
<FOR> <AP> atribuição <PTV> expressão <PTV> atribuição <FP> <ACH> listaComandos  
<FCH>

if -> <IF> <AP> expressão <FP> comando | <IF> <AP> expressão <FP> <ACH>  
listaComandos <FCH>

switch -> <SWITCH> <AP> expressão <FP> <ACH> case <FCH>  
caso -> <CASE> condicao <DPT> listaComandos | <CASE> condicao <DPT>  
listaComandos caso | lambda  
condicao -> <CARACTERE> | <NUMERO>

atribuição -> <VARIABEL> <IGUALDADE> expressão | <VARIABEL> <IGUALDADE>  
expressão <VIRG> atribuição | <VARIABEL><OP-UNARIO> | <OP-UNARIO> <VARIABEL>

expressão -> expressão <OP-BINARIO> expressão | <NUMERO> | <VARIABEL> |  
<NEGACAO> expressão | <AP> expressão <FP> | <VARIABEL><OP-UNARIO> | <OP-UNARIO>  
<VARIABEL> | <OP-UN-BIN> expressao

//////////////////////////////////TOKENS//////////////////////////////////

<WHILE> : while  
<DO> : do  
<IF> : if  
<FOR> : for  
<SWITCH> : switch  
<CASE> : case

<AP> : (  
<FP> : )  
<ACH> : {  
<FCH> : }

<PTV> : ;  
<DPT> : :

<NEGACAO> : !

<VARIABEL> : variável| sinal variável  
variável -> letra N

## GLC\_LFAv2

N -> letra | letra N | digito | digito N | lambda

letra -> a|...|z|A|...|Z

digito -> 0|...|9

<NUMERO> : inteiro | real

inteiro -> digito | digito inteiro

inteiro com sinal -> sinal digito | sinal digito inteiro

real -> inteiro . inteiro | inteiro com sinal . inteiro

sinal -> + | -

<OP-UN-BIN> : + | -

<OP-UNARIO> -> ++ | --

<OP-BINARIO> : \* | / | % | && | || | == | != | > | >= | < | <= | & | | | >> |

<<

<IGUALDADE> : = | += | -= | \*= | /= | %= | &= | |=

<CARACTERE> : ' letra '