

**Centro Universitário Norte do Espírito Santo  
Universidade Federal do Espírito Santo (UFES)**

**Linguagens Formais e Autômatos**

Ana Flavia Frontino da Cruz, Carlos Henrique Fernandes Louzada da Fonseca,  
Keicila Marcial, Julio Nico, Ramon de Souza Barbosa

**Linguagem de programação: PHP**

**TOKENS**

<IDENT> = \$ letra (letra | dígito)\*

<NUM> = (0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)+((0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9)\*)?

<WHILE> = (w | W)(h | H)(i | I)(l | L)(e | E)

<FOR> = (f | F)(o | O)(r | R)

<DO> = (d | D)(o | O)

<IF> = (i | I)(f | F)

<OP-COMP> = ==

| ===

| !=

| <>

| !==

| <

| >

| >=

| <=

<OP.BIN.LOG> = (a | A)(n | N)(d | D)

| (o | O)(r | R)

| (x | X)(o | O)(r | R)

| &&

| ||

<OP.UNI.LOG> = !

<OP-ARIT> = \*

| /

<OP.ARIT.SINAL> = +

| -  
 <ATRIB> = =  
 <AP> = (  
 <FP> = )  
 <AC> = {  
 <FC> = }  
 <PT-VIRG> = ;  
  
 letra -> ((a | A) | (b | B) | (c | C) | ... | (z | Z))+  
 digito -> (0 | 1 | 2 | ... | 9)+

## **GRAMÁTICA LIVRE DE CONTEXTO - 1º versão** **(com conflito, recursividade e ambiguidade)**

blocoComandos -> comando <PT-VIRG> blocoComandos  
 | comando <PT-VIRG>

comando -> while  
 | do  
 | for  
 | if  
 | atrib

while -> <WHILE> <AP> exp <FP> <AC> blocoComandos <FC>  
 | <WHILE> <AP> exp <FP> comando

do -> <DO> <AC> blocoComandos <FC> <WHILE> <AP> exp <FP>  
 | <DO> comando <WHILE> <AP> exp <FP>

for -> <FOR> <AP> exp <PT-VIRG> exp <PT-VIRG> exp <FP> <AC> blocoComandos  
 <FC>  
 | <FOR> <AP> exp <PT-VIRG> exp <PT-VIRG> exp <FP> comando

if -> <IF> <AP> exp <FP> <AC> blocoComandos <FC>  
 | <IF> <AP> exp <FP> comando

atrib -> <IDENT> <ATRIB> exp

exp -> exp <OP-COMP> exp  
| exp <OP.BIN.LOG> exp  
| <OP-UNI.LOG> exp  
| exp <OP-BIT-A-BIT> exp  
| exp <OP-ARIT> exp  
| exp <OP.ARIT.SINAL> exp  
| <AP> exp <FP>  
| <NUM>  
| <OP-ARIT.SINAL><NUM>  
| <IDENT>

## **GRAMÁTICA LIVRE DE CONTEXTO - 2º versão (sem conflito, recursividade e ambiguidade)**

blocoComandos -> comando <PT-VIRG> blocoComandos  
| comando <PT-VIRG>

comando -> while  
| do  
| for  
| if  
| atrib

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

do -> <DO> corpo <WHILE> <AP> exp <FP>

for -> <FOR> <AP> exp <PT-VIRG> exp <PT-VIRG> exp <FP> corpo

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

corpo -> <AC> blocoComandos <FC> | comando

atrib -> <IDENT> <ATRIB> exp

operador -> <OP-COMP>  
| <OP.BINI.LOG>  
| <OP-BIT-A-BIT>  
| <OP-ARIT>  
| <OP-ARIT.SINAL>

exp -> | <OP.UNI.LOG> exp R  
| <AP> exp R <FP>  
| <NUM> R  
| <OP-ARIT.SINAL><NUM> R  
| <IDENT> R

R -> operador exp R |  $\lambda$