**Gramática**

G = (N, T, P, prog).

**N =** {**prog, listacom, com, atrib, atrib2, vatrib, ler, cond, cond1, cond2, esc, argum, arg2, exp, expcomp, exp1, explog, expar, tipoa, tipob, tipoc, opar, opcomp**}.

**T =** {EOF, LOCAL, PALAVRA, ENDL, ATRIBUICAO, VIRGULA, IO.READ, ABREPAR, FECHAPAR, IF, ABRECHAVE, FECHACHAVE, ELSE, IO.WRITE, <STRING>, <VIRG>, TRUE, FALSE, NULL, NUM, MENOS, MAIS, NOT, DIVIZAO, MULTIPLICACAO, POTENCIA, OR, AND, MAISMENOS, RELACIONAL}.

**P =** { **prog** -> **listacom** EOF,

**listacom** -> **com listacom** | VAZIO,

**com** -> LOCAL PALAVRA **atrib** ENDL | PALAVRA **atrib2** ENDL | **cond** | **esc** ENDL,

**atrib** ->ATRIBUICAO **exp** | VAZIO,

**atrib2** ->ATRIBUICAO **ler** | **vatrib** **exp** | VAZIO,

**vatrib** -> VIRGULA PALAVRA **vatrib** **exp** VIRGULA | ATRIBUICAO,

**ler** -> **exp** | IO.READ ABREPAR FECHAPAR,

**cond** -> IF **exp** ABRECHAVE ENDL **listacom** FECHACHAVE ENDL **cond1,**

**cond1** -> **cond2** | VAZIO,

**cond2** -> ELSE ABRECHAVE ENDL **listacom** FECHACHAVE ENDL,

**esc** -> IO.WRITE ABREPAR **argum** FECHAPAR,

**argum** -> <STRING> **arg2** | **exp arg2,**

**arg2** -> <VIRG> **argum** | VAZIO,

**exp** -> **exp1** **expcomp,**

**expcomp** -> **opcomp** **exp1** | VAZIO,

**exp1** -> **expar** | **explog,**

**explog** -> TRUE | FALSE | NULL,

**expar** -> **tipoa,**

**tipoa** -> NUM **tipob** | PALAVRA **tipob** | MENOS **tipoa** | NOT **tipoa** | ABREPAR **tipoa** FECHAPAR,

**tipob** -> **opar** **tipoa** | VAZIO,

**opar** -> DIVIZAO | MULTIPLICACAO | MENOS | MAIS | POTENCIA | OR | AND,

**opcomp** -> MAISMENOS | RELACIONAL}.