

- Linguagem Lovelace, definições léxicas e sintáticas:

Linguagem Lovelace v 1.0

~~~~~

LOVELACE -> MAIN FUNC?

MAIN -> "main" "(" ")" "begin" VARDECL SEQCOMANDOS "end"

VARDECL -> VARDECL "let" TIPO TOKEN\_id ";" | vazio

TIPO -> "Float" | "Bool" | "Void"

SEQCOMANDOS -> SEQCOMANDOS COMANDO | vazio

COMANDO -> TOKEN\_id "!=" EXP ";"  
 | TOKEN\_id "(" LISTAEXP? ")" ";"  
 | "if" EXP "begin" SEQCOMANDOS "end" ";"  
 | "while" EXP "begin" SEQCOMANDOS "end" ";"  
 | TOKEN\_id "!=" "read" "(" ")" ";"  
 | "return" EXP ";"  
 | "print" EXP ";"

EXP -> "(" EXP OP EXP ")" | FATOR

FATOR -> TOKEN\_id | TOKEN\_id "(" LISTAEXP? ")"  
 | TOKEN\_numliteral | "true" | "false"

OP -> "+" | "-" | "\*" | "/" | "&&" | "||" | "<" | ">" | "=="

LISTAEXP -> EXP | LISTAEXP "," EXP

FUNC -> FUNC "def" TIPO TOKEN\_id "(" LISTAARG? ")" "begin" VARDECL SEQCOMANDOS "end"  
 | "def" TIPO TOKEN\_id "(" LISTAARG? ")" "begin" VARDECL SEQCOMANDOS "end"

LISTAARG -> TIPO TOKEN\_id | LISTAARG "," TIPO TOKEN\_id

=====

Convenções léxicas

~~~~~

TOKEN_id -> letra letraoudigito* finalsublinhado*

TOKEN_numliteral -> digitos facao_opcional expoente_opcional

onde:

letra -> [a-zA-Z]

digito -> [0-9]

digitos -> digito+

```
facao_opcional -> (.digitos)?  
expoente_opcional -> (E (+ | -)? digitos)?  
letraoudigito -> letra | digito  
finalsublinhado -> _letraoudigito+
```