- Linguagem Karloff, definições léxicas e sintáticas:

```
Linguagem Karloff
^^^^^^^^^ ^^^^^^
v 0.2

KARLOFF -> MAIN FUNC?

MAIN -> "void" "main" "(" ")" "{" VARDECL SEQCOMANDOS "}"

VARDECL -> VARDECL "newVar" TIPO TOKEN_id ";" | vazio

TIPO -> "integer" | "bool"

SEQCOMANDOS -> SEQCOMANDOS COMANDO | vazio

COMANDO ->  TOKEN_id "=" EXP ";"
   | TOKEN_id "(" LISTAEXP? ")" ";"
   | "if" "(" EXP ")" "then" "{" SEQCOMANDOS "}"  ";"
   | "while" "(" EXP ")"  "{" SEQCOMANDOS "}"  ";"
   | "repeat" "{" SEQCOMANDOS "}" "until" "(" EXP ")"  ";"
   | "return" EXP ";"
   | "System.output" "(" EXP ")"  ";"
   | TOKEN_id "=" "System.readint" "(" ")" ";"


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

FATOR -> TOKEN_id | TOKEN_id "(" LISTAEXP? ")"
         | TOKEN_numliteral |   "true" | "false"

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

LISTAEXP -> EXP | LISTAEXP "," EXP


FUNC -> FUNC "func" TIPO TOKEN_id "(" LISTAARG? ")" "{" VARDECL SEQCOMANDOS "}"
       | "func" TIPO TOKEN_id "(" LISTAARG? ")" "{" VARDECL SEQCOMANDOS "}"

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+
letra -> [a-zA-Z]
digito -> [0-9]
```