

## A linguagem Pyscal (Corrigida)

Programa → Classe EOF 1

Classe → "class" ID ":" ListaFuncao Main "end" "." 2

DeclararID → TipoPrimitivo ID ";" 3

ListaFuncao → ListaFuncao' 4

ListaFuncao' → Funcao ListaFuncao' 5 | ε 6

Funcao → "def" TipoPrimitivo ID "(" ListaArg ")" ":" RegexDeclararID ListaCmd Retorno "end" ";" 7

RegexDeclararID → DeclararID RegexDeclararID 8 | ε 9

ListaArg → Arg ListaArg' 10

ListaArg' → "," ListaArg 11 | ε 12

Arg → TipoPrimitivo ID 13

Retorno → "return" Expressao ";" 14 | ε 15

Main → "defstatic" "void" "main" "(" "String" "[" "]" ID ")" ":" RegexDeclararID ListaCmd "end" ";" 16

TipoPrimitivo → "bool" 17 | "integer" 18 | "String" 19 | "double" 20 | "void" 21

ListaCmd → ListaCmd' 22

ListaCmd' → Cmd ListaCmd' 23 | ε 24

Cmd → CmdIF 25 | CmdWhile 26 | ID CmdAtribFunc 27 | CmdWrite 28

CmdAtribFunc → CmdAtribui 29 | CmdFuncao 30

CmdIF → "if" "(" Expressao ")" ":" ListaCmd CmdIF' 31

CmdIF' → "end" ";" 32 | "else" ":" ListaCmd "end" ";" 33

CmdWhile → "while" "(" Expressao ")" ":" ListaCmd "end" ";" 34

CmdWrite → "write" "(" Expressao ")" ";" 35

CmdAtribui → "=" Expressao ";" 36

CmdFuncao → "(" RegexExp ")" ";" 37

RegexExp → Expressao RegexExp' 38 | ε 39

RegexExp' → , Expressao RegexExp' 40 | ε 41

Expressao → Exp1 Exp' 42

Exp' → "or" Exp1 Exp' 43 | "and" Exp1 Exp' 44 | ε 45

Exp1 → Exp2 Exp1' 46

Exp1' → "<" Exp2 Exp1' 47 | "<=" Exp2 Exp1' 48 | ">" Exp2 Exp1' 49 | ">=" Exp2 Exp1' 50 | "==" Exp2 Exp1' 51 | "!=" Exp2 Exp1' 52 | ε 53

Exp2 → Exp3 Exp2' 54

Exp2' → "+" Exp3 Exp2' 55 | "-" Exp3 Exp2' 56 | ε 57

Exp3 → Exp4 Exp3' 58

Exp3' → "\*" Exp4 Exp3' 59 | "/" Exp4 Exp3' 60 | ε 61

Exp4 → ID Exp4' 62 | ConstInteger 63 | ConstDouble 64 | ConstString 65 | "true" 66 | "false" 67 | OpUnario Exp4 68 | "(" Expressao ")" 69

Exp4' → "(" RegexExp ")" 70 | ε 71

OpUnario → "-" 72 | "!" 73

### Cálculo do FIRST e FOLLOW

	FRIST	FOLLOW
Programa	“class”	EOF
Classe	“class”	EOF
DeclararID	"bool", "integer", "String", "double", "void"	"bool", "integer", "String", "double", "void", "if", "while", “ID”, "write", "return", "end"
ListaFuncao	“def”, “ε”	"defstatic"
ListaFuncao’	“def”, “ε”	"defstatic"
Funcao	“def”	“def”, "defstatic"
RegexDeclararId	"bool", "integer", "String", "double", "void", “ε”	"if", "while", “ID”, "write", "return", "end"
ListaArg	"bool", "integer", "String", "double", "void"	")"
ListaArg’	“,”, “ε”	")"
Arg	"bool", "integer", "String", "double", "void"	“,”, ")"
Retorno	"return", “ε”	"end"
Main	"defstatic"	"end"
TipoPrimitivo	"bool", "integer", "String", "double", "void"	ID
ListaCmd	"if", "while", “ID”, "write", “ε”	"return", "end", "else"
ListaCmd’	"if", "while", “ID”, "write", “ε”	"return", "end", "else"
Cmd	"if", "while", “ID”, "write"	"if", "while", “ID”, "write", "return", "end", "else"
CmdAtribFunc	"=", "("	"if", "while", “ID”, "write", "return", "end", "else"
CmdIF	"if"	"if", "while", “ID”, "write", "return", "end", "else"
CmdIF’	"end", "else"	"if", "while", “ID”, "write", "return", "end", "else"
CmdWhile	"while"	"if", "while", “ID”, "write", "return", "end", "else"
CmdWrite	"write"	"if", "while", “ID”, "write", "return", "end", "else"
CmdAtribui	"="	"if", "while", “ID”, "write", "return", "end", "else"
CmdFuncao	"("	"if", "while", “ID”, "write", "return", "end", "else"
RegexExp	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "(" , “ε”	")"
RegexExp’	“,”, “ε”	")"
Expressao	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "("	")", ";", ",", "
Exp’	"or", "and", “ε”	")", ";", ",", "
Exp1	ID, ConstInteger, ConstDouble, ConstString, "true",	"or", "and", ")", ";", ",", "

	"false", "-"(negação), "!", "("	
Exp1'	"<", "<=", ">", ">=", "==", "!=", "ε"	"or", "and", ")", ";", ", "
Exp2	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "("	"<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
Exp2'	+", "-", "ε"	"<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
Exp3	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "("	+", "-", "<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
Exp3'	"*", "/", "ε"	+", "-", "<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
Exp4	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "("	"*", "/", "+", "-", "<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
Exp4'	"(", "ε"	"*", "/", "+", "-", "<", "<=", ">", ">=", "==", "!=", "or", "and", ")", ";", ", "
OpUnario	"-"(negação), "!"	ID, ConstInteger, ConstDouble, ConstString, "true", "false", "-"(negação), "!", "("