

# Adequação da gramática e classificação dos tokens

**Alunos:** Leonardo Apicelo / Gabriel Moura

## 1. Gramática revisada e adequada:

[PROGRAMA]	(program) [ID] (;) [CORPO]
[CORPO]	[DECLARACOES] (begin) [LISTA_COM] (end)
	(begin) [LISTA_COM] (end)
[DECLARACOES]	[DEF_CONST] [DEF_TIPOS] [DEF_VAR] [LISTA_FUNC]
	ε
[DEF_CONST]	(const) [LISTA_CONST]
	ε
[LISTA_CONST]	[CONSTANTE] [LISTA_CONST']
[LISTA_CONST']	[LISTA_CONST]
	ε
[CONSTANTE]	[ID] (:=) [CONST_VALOR] (;)
[CONST_VALOR]	(") sequência alfanumérica (")
	[EXP_MATEMATICA]
[DEF_TIPOS]	(type) [LISTA_TIPOS]
	ε

[LISTA_TIPOS]	[TIPO] [LISTA_TIPOS']
[LISTA_TIPOS']	(;) [LISTA_TIPOS]
	€
[TIPO]	[ID] (:=) [TIPO_DADO]
[TIPO_DADO]	(integer)
	(real)
	(array) ([] [NUMERO] []) (of) [TIPO_DADO]
	(record) [LISTA_VAR] (end)
	[ID]
[DEF_VAR]	(var) [LISTA_VAR]
	€
[LISTA_VAR]	[VARIABEL] [LISTA_VAR']
[LISTA_VAR']	(;) [LISTA_VAR]
	€
[VARIABEL]	[LISTA_ID] (:) [TIPO_DADO]
[LISTA_ID]	[ID] [LISTA_ID']
[LISTA_ID']	(,) [LISTA_ID]
	€
[NUMERO]	Sequência de dígitos com a ocorrência de no máximo um ponto (tratado no léxico)
[LISTA_FUNC]	[FUNCAO] [LISTA_FUNC]
	€

[FUNCAO]	[NOME_FUNCAO] [BLOCO_FUNCAO]
[NOME_FUNCAO]	(function) [ID] (() [LIST_VAR] ()) (:) [TIPO_DADO]

[BLOCO_FUNCAO]	[DEF_VAR] [BLOCO]
	[BLOCO]
[BLOCO]	(begin) [LISTA_COM] (end)
	[COMANDO]
[LISTA_COM]	[COMANDO] (;) [LISTA_COM]
	ε
[COMANDO]	[NOME] (:=) [VALOR]
	(while) [EXP_LOGICA] [BLOCO]
	(if) [EXP_LOGICA] (then) [BLOCO] [ELSE]
	(write) [CONST_VALOR]
	(read) [NOME]
[ELSE]	(else) [BLOCO]
	ε
[VALOR]	[EXP_MAT]
	[ID] [LISTA_PARAM]
[LISTA_PARAM]	(()) [LISTA_NOME] (())
[LISTA_NOME]	[PARAMETRO] [LISTA_NOME']
	ε
[LISTA_NOME']	(,) [LISTA_NOME]

	$\epsilon$
[EXP_LOGICA]	[EXP_MAT] [EXP_LOGICA']
[EXP_LOGICA']	[OP_LOGICO] [EXP_LOGICA]
	$\epsilon$
[EXP_MAT]	[PARAMETRO] [EXP_MAT']

[EXP_MAT']	[OP_MAT] [EXP_MAT]
	$\epsilon$
[PARAMETRO]	[NOME]
	[NUMERO]
[OP_LOGICO]	(>)   (<)   (=)   (!)
[OP_MAT]	(+)   (-)   (*)   (/)
[NOME]	[ID] [NOME']
[NOME']	. [NOME]
	([PARAMETRO])
	$\epsilon$
[ID]	Sequência alfanumérica iniciada por char (tratado no léxico)

## 2. Classificação dos tokens:

Token	Padrão
ID	Sequência alfanumérica iniciada por char (tratado no léxico)

Num	Sequência de dígitos com a ocorrência de no máximo um ponto (tratado no léxico)
Prog	Palavra reservada "program"
PontV	;
Begin	Palavra reservada "begin"
End	Palavra reservada "end"

Const	Palavra reservada "const"
OpLog	= ou ! ou < ou >
OpMat	+ ou - ou * ou /
Aspas	"
Type	Palavra reservada "type"
TipoSimples	São as palavras reservadas: "real" e "integer"
Array	Palavra reservada "array"
Record	Palavra reservada "record"

AColch	[
FColch	]
Of	Palavra reservada “of”
Var	Palavra reservada “var”
DoisPt	:
Virg	,
Func	Palavra reservada “function”
AParent	(

FParent	)
Atribuição	Sequência de símbolos :=
Loop	Palavra reservada “while”
Se	Palavra reservada “if”
Então	Palavra reservada “then”
Senão	Palavra reservada “else”

Escrita	Palavra reservada "write"
Leitura	Palavra reservada "read"
Erro	Nenhum padrão reconhecido

Observações:

- Os tokens não estão em ordem de prioridade.
- Vamos considerar # como marcador de início e fim de comentário.