

Lucas Peixoto de Almeida Cavalcante

## **Especificação da gramática da linguagem de programação LisC**

Especificação da gramática da linguagem de programação LisC, definida pelo aluno, para a disciplina de Compiladores, correspondente à parte da avaliação da AB2 do semestre de 2019.1, sob orientação do **Prof. Alcino Dall Igna Jr.**

Universidade Federal de Alagoas

Instituto de Computação

Brasil

Maceió/Al  
2019.1

# Sumário

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>3</b>
<b>2</b>	<b>GRAMÁTICA ORIGINAL . . . . .</b>	<b>4</b>
<b>3</b>	<b>GRAMÁTICA LL(1) . . . . .</b>	<b>7</b>

# 1 Introdução

Esse documento descreve a gramática construída da linguagem de programação LisC, que deverá ser respeitada na implementação do analisador sintático em C. O analisador sintático escolhido é o preditivo recursivo. Diante da escolha deste analisador sintático, teremos também nesse documento as alterações necessárias na gramática para que a gramática se torne uma LL(1), pré-requisito obrigatório de todo analisador sintático preditivo recursivo.

## 2 Gramática Original

O código 3.1 apresenta a gramática original da linguagem.

```

1 Programa = ListaDecl
2
3 ListaDecl = ListaDecl Decl
4 ListaDecl = Decl
5
6 Decl = DeclVariaveis
7 Decl = DeclFuncao
8
9 DeclVariaveis = DefTipo ListaVariaveis ';'
10
11 DefTipo = 'defint'
12 DefTipo = 'defbool'
13 DefTipo = 'defchar'
14 DefTipo = 'defstring'
15 DefTipo = 'deffloat'
16
17 ListaVariaveis = ListaVariaveis ',' Variavel
18 ListaVariaveis = Variavel
19
20 Variavel = 'id'
21 Variavel = 'id' ExpConcat
22
23 DeclFuncao = 'defun' 'id' '(' ListaParam ')' '(' TipoReturn ')' '{'
    ListaSent '}'
24
25 ListaParam = ListaParam ',' Param
26 ListaParam = Param
27
28 Param = Tipo 'id'
29
30 TipoReturn = 'void'
31 TipoReturn = Tipo
32
33 ListaSent = ListaSent Sent
34 ListaSent = Sent
35
36 Sent = DeclVariaveis
37 Sent = Comando
38
39 Comando = Continue
40 Comando = Break

```

```

41 Comando = Atribuicao
42 Comando = ChamadaFunc
43 Comando = Return
44 Comando = Read
45 Comando = Echo
46 Comando = If
47 Comando = For
48 Comando = While
49
50 Continue = 'continue' ';'
51
52 Break = 'break' ';'
53
54 Atribuicao = 'set' 'id' ExpConcat ';'
55
56 ChamadaFunc = 'id' '(' ListaArg ')' ';'
57
58 Return = 'return' ExpConcat ';'
59
60 Read = 'read' '(' ListaArg ')' ';'
61
62 Echo = 'echo' '(' ExpConcat ')'
63 Echo = 'echo' '(' ExpConcat ')' '(' ListaArg ')'
64
65 If = 'if' '(' ExpBooleana ')' '{' ListaSent '}' Else
66
67 Else = 'else' '{' ListaSent '}'
68 Else = epsilon
69
70 For = 'for' '(' 'id' ',' ExpAritmetica ',' ExpAritmetica ForPasso ')' '{'
    ListaSent '}'
71
72 While = 'while' '(' ExpBooleana ')' '{' ListaSent '}'
73
74 ForPasso = ',' ExpAritmetica
75 ForPasso = epsilon
76
77 ListaArg = ListaArg ',' 'id'
78 ListaArg = 'id'
79
80 Tipo = 'int'
81 Tipo = 'char'
82 Tipo = 'string'
83 Tipo = 'float'
84 Tipo = 'bool'
85
86 ExpAritmetica = ExpAritmetica 'op_adi' TermAritmetico

```

```
87 ExpAritmetica = TermAritmetico
88
89 ExpBooleana = ExpBooleana 'op_logic' TermBooleano
90 ExpBooleana = TermBooleano
91
92 ExpConcat = ExpConcat '<<' ExpAritmetica
93 ExpConcat = ExpBooleana
94
95 TermBooleano = '!' TermBooleano
96 TermBooleano = ExpAritmetica 'op_relacional' ExpAritmetica
97 TermBooleano = ExpAritmetica
98
99 TermAritmetico = TermAritmetico 'op_mult_div' TermFormatacao
100 TermAritmetico = TermFormatacao
101
102 TermFormatacao = TermFormatacao 'op_format' FatorAritmetico
103 TermFormatacao = FatorAritmetico
104
105 FatorAritmetico = '-' FatorAritmetico
106 FatorAritmetico = 'id'
107 FatorAritmetico = 'const_float'
108 FatorAritmetico = 'const_char'
109 FatorAritmetico = 'const_bool'
110 FatorAritmetico = 'const_int'
111 FatorAritmetico = 'const_string'
112 FatorAritmetico = '(' ExpBooleana ')'
```

Código 2.1 – Gramática original da linguagem de programação LisC.

## 3 Gramática LL(1)

```
1 Programa = ListaDecl
2
3 ListaDecl = Decl ListaDeclR
4
5 ListaDeclR = Decl ListaDeclR
6 ListaDeclR = epsilon
7
8 Decl = DeclVariaveis
9 Decl = DeclFuncao
10
11 DeclVariaveis = DefTipo ListaVariaveis ';'
12
13 DefTipo = 'defint'
14 DefTipo = 'defbool'
15 DefTipo = 'defchar'
16 DefTipo = 'defstring'
17 DefTipo = 'deffloat'
18
19 ListaVariaveis = Variavel ListaVariaveisR
20
21 ListaVariaveisR = ',' Variavel ListaVariaveisR
22 ListaVariaveisR = epsilon
23
24 Variavel = 'id' VariavelF
25
26 VariavelF = ExpConcat
27 VariavelF = epsilon
28
29 DeclFuncao = 'defun' 'id' '(' ListaParam ')' '(' TipoReturn ')' '{'
    ListaSent '}'
30
31 ListaParam = Param ListaParamR
32
33 ListaParamR = ',' Param ListaParamR
34 ListaParamR = epsilon
35
36 Param = Tipo 'id'
37
38 TipoReturn = 'void'
39 TipoReturn = Tipo
40
41 ListaSent = Sent ListaSentR
42
```

```
43 ListaSentR = Sent ListaSentR
44 ListaSentR = epsilon
45
46 Sent = DeclVariaveis
47 Sent = Comando
48
49 Comando = Continue
50 Comando = Break
51 Comando = Atribuicao
52 Comando = ChamadaFunc
53 Comando = Return
54 Comando = Read
55 Comando = Echo
56 Comando = If
57 Comando = For
58 Comando = While
59
60 Continue = 'continue' ';'
61
62 Break = 'break' ';'
63
64 Atribuicao = 'set' 'id' ExpConcat ';'
65
66 ChamadaFunc = 'id' '(' ListaArg ')' ';'
67
68 Return = 'return' ExpConcat ';'
69
70 Read = 'read' '(' ListaArg ')' ';'
71
72 Echo = 'echo' '(' ExpConcat ')' EchoF
73
74 EchoF = '(' ListaArg ')' ';'
75 EchoF = ';'
76
77 If = 'if' '(' ExpBooleana ')' '{' ListaSent '}' Else
78
79 Else = 'else' '{' ListaSent '}'
80 Else = epsilon
81
82 For = 'for' '(' 'id' ',' ExpAritmetica ',' ExpAritmetica ForPasso ')' '{'
    ListaSent '}'
83
84 While = 'while' '(' ExpBooleana ')' '{' ListaSent '}'
85
86 ForPasso = ',' ExpAritmetica
87 ForPasso = epsilon
88
```



```

89 ListaArg = 'id' ListaArgR
90
91 ListaArgR = ',' 'id' ListaArgR
92 ListaArgR = epsilon
93
94 Tipo = 'int'
95 Tipo = 'char'
96 Tipo = 'string'
97 Tipo = 'float'
98 Tipo = 'bool'
99
100 ExpAritmetica = TermAritmetico ExpAritmeticaR
101
102 ExpAritmeticaR = 'op_adi' TermAritmetico ExpAritmeticaR
103 ExpAritmeticaR = epsilon
104
105 ExpBooleana = TermBooleano ExpBooleanaR
106
107 ExpBooleanaR = 'op_logic' TermBooleano ExpBooleanaR
108 ExpBooleanaR = epsilon
109
110 ExpConcat = ExpBooleana ExpConcatR
111
112 ExpConcatR = '<<' ExpAritmetica ExpConcatR
113 ExpConcatR = epsilon
114
115 TermBooleano = '!' TermBooleano
116 TermBooleano = ExpAritmetica TermBooleanoF
117
118 TermBooleanoF = 'op_relacional' ExpAritmetica
119 TermBooleanoF = epsilon
120
121 TermAritmetico = TermFormatacao TermAritmeticoR
122
123 TermAritmeticoR = 'op_mult_div' TermFormatacao TermFormatacaoR
124 TermAritmeticoR = epsilon
125
126 TermFormatacao = FatorAritmetico TermFormatacaoR
127
128 TermFormatacaoR = 'op_format' FatorAritmetico TermFormatacaoR
129 TermFormatacaoR = epsilon
130
131 FatorAritmetico = '-' FatorAritmetico
132 FatorAritmetico = 'id'
133 FatorAritmetico = 'const_float'
134 FatorAritmetico = 'const_char'
135 FatorAritmetico = 'const_bool'

```

```
136 FatorAritmetico = 'const_int'  
137 FatorAritmetico = 'const_string'  
138 FatorAritmetico = '(' ExpBooleana ')'
```

Código 3.1 – Gramática original da linguagem de programação LisC.