

```
1 (...)
2 declaracao_variavel
3   : tipo lista_variaveis declaracao_variavel
4   | tipo lista_variaveis
5   ;
6
7 tipo
8   : T_INTEIRO
9   | T_LOGICO
10  ;
11
12 // versao original adaptada para aceitar declaracao de vetor
13 lista_variaveis
14   : lista_variaveis T_IDENTIF { cadastrar a variavel }
15   | T_IDENTIF { cadastrar a variavel }
16   | lista_variaveis T_IDENTIF T_ABREC T_NUMERO T_FECHAC { cadastrar a
variavel }
17   | T_IDENTIF T_ABREC T_NUMERO T_FECHAC { cadastrar a variavel }
18   ;
19
20 // sugestao da modificacao da gramatica para aceitar vetor
21 lista_variaveis
22   : lista_variaveis variavel
23   | variavel
24   ;
25
26 variavel
27   : T_IDENTIF
28     { strcpy (nome, atomo); }
29     tamanho
30   ;
31
32 tamanho
33   : /* vazio */ { cadastrar uma variavel simples }
34   | T_ABREC T_NUMERO { cadastrar um vetor } T_FECHAC
35   ;
36
37
38 // Onde mais aparece o vetor:
39 // 1. No lado esquerdo de uma atribuicao
40
41 atribuicao
42   : T_IDENTIF
43     {
44       int pos = busca_simbolo(atomo);
45       if (pos == -1)
46         erro ("Variavel não declarada!");
47       empilha (pos);
48     }
49     posicao T_ATRIB expr
50     {
51       int tipo = desempilha();
52       int pos = desempilha ();
53       gerar um ARZG ou ARZV
54     }
55   ;
56
57 atribuicao
58   : T_IDENTIF T_ATRIB expr;
```

```

59 | T_IDENTIF T_ABREC expr T_FECHAC T_ATRIB expr
60 ;
61
62 /*-----
63 Alguns exemplos de tradução para atribuicao de vetores e variaveis
64
65 vetor[i*4 + j] <- 2*i
66   CRVG i
67   CRCT 4
68   MULT
69   CRVG j
70   SOMA  -----> M[s-1]
71
72   CRCT 2
73   CRVG i
74   MULT  -----> M[s]
75
76   ARZV n -----> M[n + M[s-1]] <- M[s]
77               s <- s - 1
78
79 x <- vetor[x + 1]
80   CRVG x
81   CRCT 1
82   SOMA  -----> M[s]
83
84   CRVV n -----> M[s] <- M[n + M[s]]
85
86 -----*/
87
88 posicao
89 : /*vazia*/
90 | T_ABREC expr
91   {
92     int t = desempilha();
93     int p = desempilha();
94     if (t == 'l')
95       erro ("tipo do indice deve ser inteiro");
96     if (TabSimb[p].cat != VET)
97       erro ("variavel nao eh vetor.");
98     empilha (p);
99   }
100   T_FECHAC
101 ;
102
103
104
105 // 2. Numa expressao
106 (...)
107 (...)
108
109 termo
110 : T_IDENTIF indice
111 | T_NUMERO
112 (...)
113
114 indice
115 : /*vazia*/
116 | T_ABREC expr T_FECHAC
117 ;
118

```

```

119 /*-----
120 inteiro b
121 vetor[vetor[i+1]] <- 5b[10] <- 5
122
123 a[i + 10] <- i + 10
124
125     CRVG i
126     CRCT 10
127     SOMA
128     CRVG i
129     CRCT 10
130     SOMA
131     ARZV a
132 -----*/
133
134 // 3. Na leitura também.
135
136 leitura
137   : T_LEIA T_IDENTIF
138     { inclua aqui uma ação para buscar e empilha na pilha semantica a
139       posição do identificador reconhecido }
139     posicao // ou outro nome para essa variável na gramática, se vc já usou
140     posicao em alguma regra
141   ;
142 posicao
143   : /* vazio */
144     { inclua aqui uma ação para desempilhar a posição da variável na tabela
145       de símbolos e gerar LEIA e ARZG
146     | T_ABREC expr T_FECHAC
147       { inclua aqui um ação para desempilhar o tipo da expressão e a posição
148         para verificar se a variável é vetor,
149         se a expr é do tipo inteiro e então gerar o LEI e ARZV apropriado }
150   ;
151 /*-----
152 leia i
153 vetor[i] <- 5
154
155 leia x
156 vetor[0] <- x // supondo que o endereco do vetor 0
157
158     LEIA
159     ARZG x
160
161     CRCT 0 -----> M[s-1]
162     CRVG x -----> M[s]
163
164     ARZV 0 -----> M[n + M[s-1]] <- M[s]
165               s <- s - 1
166
167 leia vetor [x + 1]
168     CRVG x
169     CRCT 1
170     SOMA -----> M[s-1]
171
172     LEIA -----> M[s]
173
174     ARZV 0 -----> M[n + M[s-1]] <- M[s]
175               s <- s - 1
176 -----*/

```