

Teórico 12

Gustavo Lopes Rodrigues

3 de Junho de 2020

1 Doidona

```
1 class Doidona {
2     final int TAMT1 = 100;
3     final int TAMT3 = 100;
4     final int NULO = -0x7FFFFFFF;
5
6     int[] t1;
7     int[] t3;
8
9     ArvoreBinaria arvoreBinaria;
10    ListaSimples lista;
11    AVL arvoreAVL;
12
13    public Doidona(){
14        t1 = new int [TAMT1];
15        t3 = new int [TAMT3];
16
17        for(int i = 0; i < TAMT1; i++){
18            t1[i] = NULO;
19        }
20        for(int i = 0; i < TAMT3; i++){
21            t3[i] = NULO;
22        }
23
24        arvoreBinaria = new ArvoreBinaria();
25        arvoreAVL = new AVL();
```

```

26     lista = new ListaSimples();
27 }
28
29 public void inserir(int elemento){
30     int h = hashT1(elemento);
31     if(elemento == NULO) {
32         System.out.println("Erro! O elemento que estah tentando
inserir eh NULO");
33     } else if(t1[h] == NULO){
34         t1[h] = elemento;
35     }else if(hashT2(elemento) == 0){
36         int h = hashT3(elemento);
37
38         if(t3[h] == NULO){
39             t3[h] = elemento;
40         } else {
41             h = rehashT3(elemento);
42
43             if(t3[h] == NULO){
44                 t3[h] = elemento;
45             } else {
46                 arvoreBinaria.inserir(elemento);
47             }
48         }
49     }else if (hashT2(elemento) == 1){
50         lista.inserirFim(elemento);
51     }else if (hashT2(elemento) == 2){
52         arvoreAVL.inserir(elemento);
53     } else {
54         System.out.println("Erro!");
55     }
56 }
57 public void remover (int valor){
58
59     if(elemento == NULO) {
60         System.out.println("Erro! O elemento que estah tentando
inserir eh NULO");
61     }

```

```

62     else {
63         int h = hashT1(valor);
64         if ( t1[h] == valor ) {
65             t1[h] = NULO;
66         }
67         if ( hashT2(valor) == 0 ) {
68             h = hashT3(valor);
69             if ( t3[h] == valor ) {
70                 t3[h] = NULO;
71             }
72             else {
73                 h = rehashT3(valor);
74                 if ( t3[h] == valor ) {
75                     t3[h] = NULO;
76                 }
77                 else if ( arvoreBinaria.pesquisar(valor) == true ) {
78                     arvoreBinaria.remover(valor);
79                 }
80             }
81         }
82         else if ( hashT2(valor) == 1 && lista.pesquisar(valor) ==
true ) {
83             lista.remover(valor);
84         }
85         else if ( hashT2(valor) == 2 && arvoreAVL.pesquisar(valor) ==
true ) {
86             arvoreAVL.remover(valor);
87         }
88         else {
89             System.out.println("Erro! Esse elemento nao estah na
estrutura!");
90         }
91     }
92 }
93
94 public boolean pesquisar (int valor){
95
96     boolean resp = false;

```

```

97     int h = hashT1(valor);
98     if(t1[h] == valor){
99         resp = true;
100    }else {
101        h = hashT2(valor);
102        if (h == 0){
103            h = hashT3(valor);
104            if(t3[h] == valor){
105                resp = true;
106            }else{
107                h = rehashT3(valor);
108                if(t3[h] == valor){
109                    resp = true;
110                }else{
111                    resp = arvoreBinaria.pesquisar(valor);
112                }
113            }
114        }else if (h == 1){
115            resp = lista.pesquisar(valor);
116        } else {
117            resp = arvoreAVL.pesquisar(valor);
118        }
119    }
120    return resp;
121 }
122
123 }

```