

```

1: public class TabelaHash {
2:     //Estrutura
3:     private HumanList[] tableNOME;
4:     private HumanList[] tableCPF;
5:     //Construtor
6:     public TabelaHash (int x) {
7:         tableNOME = new HumanList[x];
8:         tableCPF = new HumanList[x];
9:
10:        for (int i=0; i<x; i++) {
11:            tableNOME[i] = new HumanList();
12:            tableCPF[i] = new HumanList();
13:        }
14:    }
15:    /** Insere nodos na HashTable a partir de seus dados */
16:    public void insertRecord (
17:
18:        String myName,
19:        String myAdress,
20:        String myCpf,
21:        char mySexo,
22:        int myAge ) {
23:
24:        HumanNode record = new HumanNode ( myName, myAdress, myCpf, mySexo, myAge );
25:
26:        int a = hashCodigo(record.getName());
27:        tableNOME[a].insere (record);
28:
29:        int b = hashCodigo(record.getCpf());
30:        tableCPF[b].insere (record);
31:    }
32:    /** Busca na tabela todos os registros com o nome especificado */
33:    public HumanList getRegistroNome ( String name ) {
34:        int a = hashCodigo(name);
35:        return tableNOME[a].getByName ( name );
36:    }
37:    /** Busca na tabela um registro com o CPF especificado */
38:    public HumanNode getRegistroCPF ( String cpf ) {
39:        int b = hashCodigo(cpf);
40:        return tableCPF[b].getByCPF ( cpf );
41:    }
42:    /** Gera um codigo hash para uma String que se encaixe na tabela */
43:    public static int hashCodigo (String stri) {
44:
45:        String str = stri.toLowerCase();
46:
47:        int primo1 = 11;
48:        int primo2 = 13;
49:
50:        long hash = 0;
51:
52:        for ( int i = 0; i < str.length(); i++ ) {
53:            hash += primo1 * str.charAt(i);
54:            primo1 *= primo2;
55:        }
56:        if ( hash >= 0 ) return (int) ( hash % 13 );
57:        else return (int) -( hash % 13 );
58:    }
59: }

```