

```
1 #include "bintree.h"
2 #include <iostream>
3
4 #include <cmath>
5 #include <queue>
6 #include<iomanip>
7
8 using namespace std;
9
10 template <class T>
11 void MostrarArbol(const bintree<T> &A,typename bintree<T>::node root){
12     queue<typename bintree<T>::node> colaNodos;
13     int totalNodos=A.size();
14     int techo=log2(totalNodos+1);
15     colaNodos.push(root);
16     int pot=0;
17     while(colaNodos.size() > 0){
18         int niveles = colaNodos.size();
19         while(niveles > 0){
20             typename bintree<T>::node nodoAux = colaNodos.front();
21             colaNodos.pop();
22             cout<<setw((niveles==pow(2,pot))?pow(2, (techo-pot)):pow(2, (techo-
pot+1))));
23             cout<<*nodoAux;
24             if(!nodoAux.left().null()) colaNodos.push(nodoAux.left());
25             if(!nodoAux.right().null()) colaNodos.push(nodoAux.right());
26             niveles--;
27         }
28         pot++;
29         cout << endl;
30     }
31 }
32
33 template <class T>
34 bool esHoja(const bintree<T> &A, const typename bintree<T>::node &v)
35 {
36     return ( v.left().null() && v.right().null() );
37 }
38
39 template <class T>
40 int profundidad(const bintree<T> &A, const typename bintree<T>::node &v)
41 {
42     int prof=0;
43     typename bintree<T>::node aux=v;
44     while(A.root()!=aux){
45         prof++;
46         aux=aux.parent();
47     }
48     return prof;
49 }
50
51 template <class T>
52 int densidad(const bintree<T> &A, typename bintree<T>::node v){
53     static int suma=0;
54
55     if(v==A.root())
56     {
57         suma=0;
58     }
59 }
```

```
60     if(!v.null())
61     {
62         if(esHoja(A,v))
63         {
64             suma+=profundidad(A,v);
65         }
66         densidad(A,v.left());
67         densidad(A,v.right());
68     }
69
70     return suma;
71 }
72
73 int main()
74 {
75     bintree<int> arb(0);
76     arb.insert_left(arb.root(),1);
77     arb.insert_right(arb.root(),2);
78
79     bintree<int>::node aux = arb.root().left();
80     arb.insert_left(aux,3);
81     arb.insert_left(aux.left(),3);
82     arb.insert_right(aux,4);
83
84     aux = arb.root().right();
85     arb.insert_left(aux,5);
86     arb.insert_right(aux,6);
87
88     MostrarArbol(arb,arb.root());
89
90     cout << "La densidad del arbol es -> " << densidad(arb,arb.root()) << endl;
91 }
```