

CAPTURAS DE PANTALLA

ARBOL BINARIO

```
arbolBinario.cpp
1  #include <iostream>
2  #include <cstdlib>
3  #include <string>
4
5  using namespace std;
6
7  //RAQUEL ALEJANDRA RUIZ RIVAS -RR106117
8
9  //permite crear un tipo de datos
10 struct node {
11     int key_value;
12     node *left;
13     node *right;
14 };
15
16 class btrees{
17
18     public:
19         btrees();
20         ~btrees();
21         void insert(int key);
22         node *search(int key);
23         void destroy_tree();
24         node *root_retriever();
25         void mostrarArbol(string prefijo, node *arbol, bool ladoDerecho);
26         void imprimirArbol();
27
28     private:
29         void destroy_tree(node *leaf);
30         void insert(int key, node *leaf);
31         node *search(int key, node *leaf);
32         node *root; //puntero tipo nodo root
33
34 };
35
```

F:\ciclo 02-2021\ESTRUCTURA DE DATOS\UNIDAD 4\EJERCICIOS\arbolBinario.exe

```
--12
|--4
|   |--3
|   |   |--1
|   |   |--9
|   |       |--7
|--18
|   |--16
-----
Process exited after 1.149 seconds with return value 0
```

```

arbolBinario.cpp  arboles binarios- RR106117.cpp
1  #include<iostream>
2  #include<cstdlib>
3
4  using namespace std;
5
6  struct node
7  {
8      int key_value;
9      node *left;
10     node *right;
11 };
12
13 class btree
14 {
15     public:
16         btree();
17         ~btree();
18         void insert(int key);
19         node *search(int key);
20         void destroy_tree ();
21         node *root_retriever();
22
23     private:
24         void destroy_tree(node *leaf);
25         void insert(int key, node *leaf);
26         node *search(int key,node *leaf);
27         node *root;
28 };
29
30 btree::btree()
31 {
32     root =NULL ;
33 }
34
35 btree::~~btree()

```

F:\ciclo 02-2021\ESTRUCTURA DE DATOS\UNIDAD 4\EJERCICIOS\arboles binarios- RR106117.exe

```

( 10)
/  \
(r) (r)

-----
Process exited after 3.827 seconds with return value 3221225477
Presione una tecla para continuar . . .

```