

```
typedef struct nodoA {
    T info;
    nodoA * Hlq;
    nodoA * HDer;
} nodo;
typedef nodo * Arbol;
```

Implementación de Operaciones Básicas

La siguiente es la implementación de las operaciones antes mencionadas, sobre un árbol binario definido como en los ejemplos anteriores, de tipo base T.

```
/* crear un árbol vacío */
void Crear (Arbol &a)
{
    a = NULL;
}
/* saber si el árbol está vacío */
boolean EsVacio(Arbol a)
{
    return (boolean) (a == NULL);
}
/* devolver la raíz del árbol */
/* Precondición: Arbol NO vacío */
T DarRaiz (Arbol a)
{
    return a->info;
}
/* obtener el subárbol izquierdo */
/* Precondición: Arbol NO vacío */
Arbol HijoIzq(Arbol a)
{
    return a->Hlq;
}
/* obtener el subárbol derecho */
/* Precondición: Arbol NO vacío */
Arbol HijoDer(Arbol a)
{
    return a->HDer;
}

/* dados dos árboles y un valor, devolver un nuevo árbol
colocando dicho valor como una nueva raíz y a los dos
árboles como subárboles de la misma */
Arbol Cons(T r,Arbol i,Arbol d)
{
    Arbol a = new nodo;
    a->info = r;
    a->Hlq = i;
    a->HDer = d;
    return a;
}
```