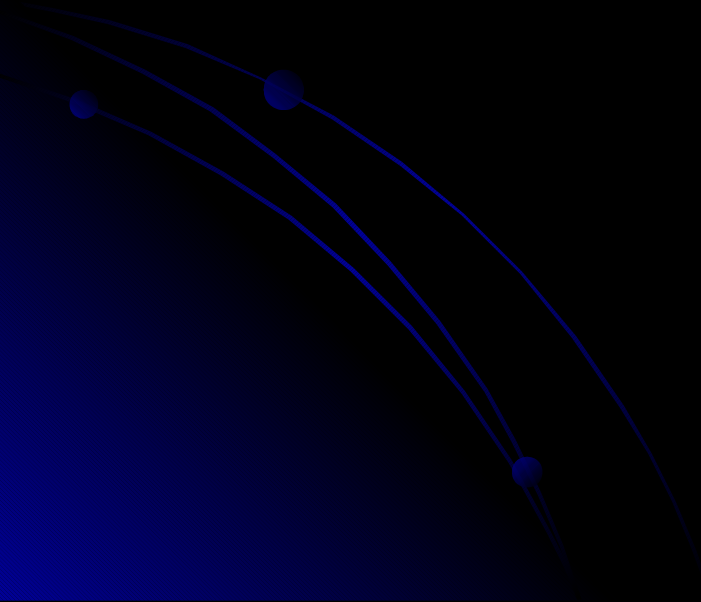


Árbol binario



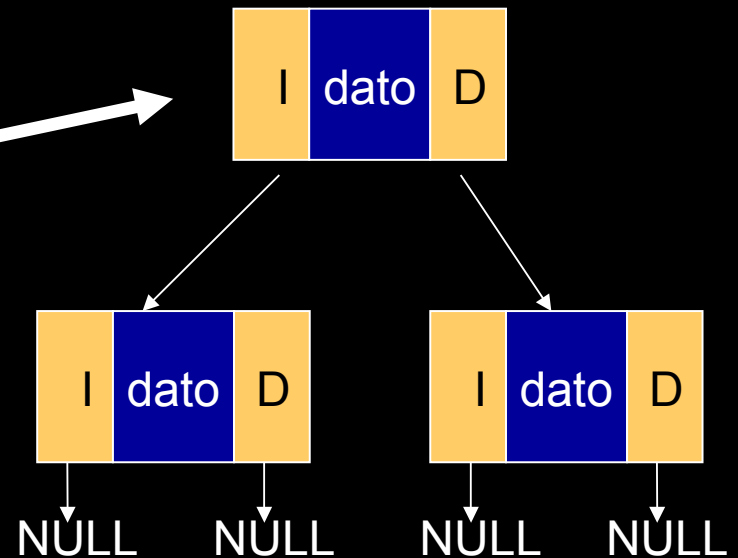
Definición

Un Árbol Binario es un conjunto finito de Elementos, de nombre Nodos de forma que:

- El Árbol Binario es Vacio si no tiene ningún elemento en el.
- El Árbol Binario contiene un Nodo Raíz y dos que parten de él, llamados Nodo Izquierdo y Nodo Derecho (que también son árboles).

Estructura de datos

```
struct nodoArbol  
{  
    int dato;  
    nodoArbol *izq;  
    nodoArbol *der;  
};
```

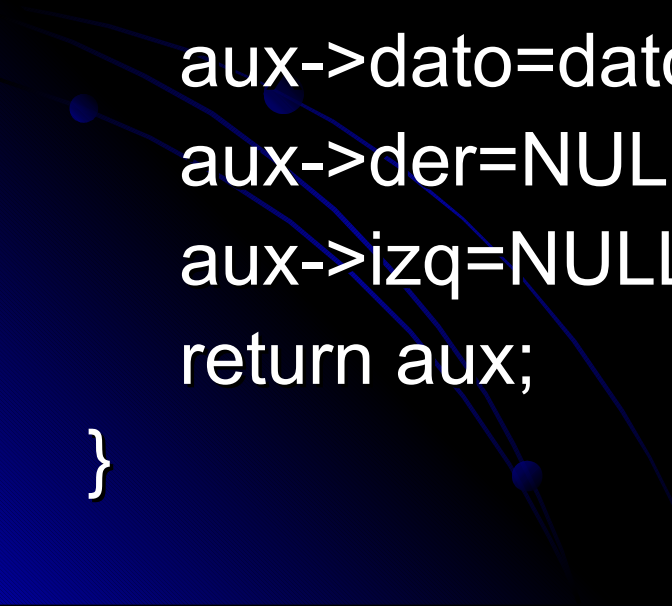


Árbol formado por un nodo raíz y dos subárboles


```
nodoArbol * inicArbol()
{
    return NULL;
}

nodoArbol * crearNodoArbol(int dato)
{
    nodoArbol * aux = (nodoArbol *) malloc(sizeof
                                                ( nodoArbol ) );

    aux->dato=dato;
    aux->der=NULL;
    aux->izq=NULL;
    return aux;
}
```

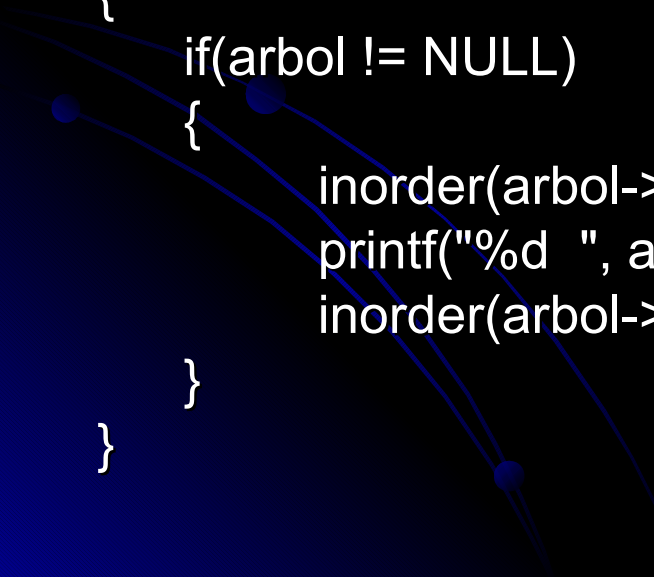


```
nodoArbol * insertar(nodoArbol * arbol, int dato)
{
    if(arbol==NULL)
        arbol = crearNodoArbol(dato);
    else
    {
        if(dato>arbol->dato)
            arbol->der = insertar(arbol->der, dato);
        else
            arbol->izq = insertar(arbol->izq, dato);
    }
    return arbol;
}
```

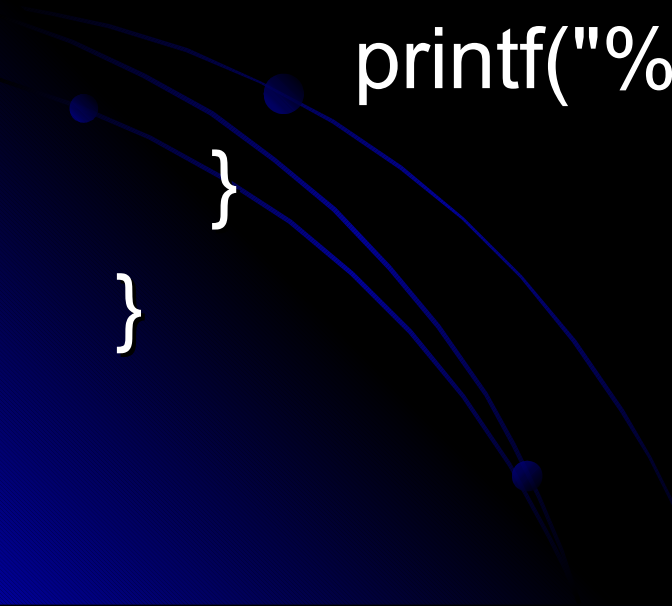


```
void preorder(nodoArbol * arbol)
{
    if(arbol != NULL)
    {
        printf("%d ", arbol->dato);
        preorder(arbol->izq);
        preorder(arbol->der);
    }
}
```

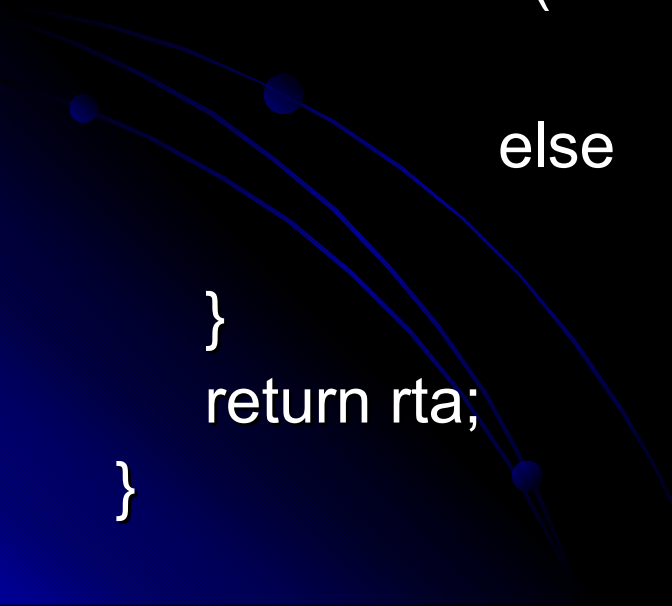
```
void inorder(nodoArbol * arbol)
{
    if(arbol != NULL)
    {
        inorder(arbol->izq);
        printf("%d ", arbol->dato);
        inorder(arbol->der);
    }
}
```



```
void postorder(nodoArbol * arbol)
{
    if(arbol != NULL)
    {
        postorder(arbol->izq);
        postorder(arbol->der);
        printf("%d ", arbol->dato);
    }
}
```



```
nodoArbol * buscar(nodoArbol * arbol, int dato)
{
    nodoArbol * rta=NULL;
    if(arbol!=NULL)
    {
        if(dato == arbol->dato)
            rta = arbol;
        else
            if(dato>arbol->dato)
                rta = buscar(arbol->der, dato);
            else
                rta = buscar(arbol->izq, dato);
    }
    return rta;
}
```




```
void main()
{
    int a[10]= {1,10,2,45,9,15,46,33,25,20};
    nodoArbol * arbol = inicArbol();
    for (int i=0; i<10; i++)
    {
        arbol = insertar(arbol, a[i]);
    }
    printf("PREORDER :");
    preorder(arbol);
    printf("\nINORDER :");
    inorder(arbol);
    printf("\nPOSTORDER :");
    postorder(arbol);
}
```