

1.- ESPECIFICACIÓN:

tad ArbolBinario(telemento) { Arbol está formado por elementos de tipo telemento}
género

arbolBin

Especificación de las operaciones:

accion iniciarArbol(**sal** A : arbolBin)
{ Inicia A como un árbol vacío }

accion enraizar(**sal** A : arbolBin, **E/S** Aiz : arbolBin,
 E/S Ade : arbolBin, **ent** d : telemento)
{ Construye el árbol A en cuya raíz se sitúa el dato d
 del cual penden los árboles Aiz y Ade. El acceso a
 los subárboles de A mediante Aiz y Ade queda anulado }

funcion izquierdo(A : arbolBin) **devuelve** arbolBin
{ Devuelve el árbol izquierdo que pende del nodo raíz
 de A }

funcion derecho(A : arbolBin) **devuelve** arbolBin
{ Devuelve el árbol derecho que pende del nodo raíz
 de A }

funcion raiz(A : arbolBin) **devuelve** telemento
{ Devuelve el dato situado como nodo raíz del árbol A }

funcion arbolVacio(A : arbolBin) **devuelve** booleano
{ Si A está vacío devuelve VERDAD y FALSO en otro caso }

2.- IMPLEMENTACIÓN DINÁMICA:

• **REPRESENTACIÓN:**

```
tipo
    Celda = registro
        dato : telemento
        izdo : puntero a Celda
        dcho : puntero a Celda
    freg;

    arbolBin = puntero a Celda
```

• **INTERPRETACIÓN:**

arbolBin es un puntero a una celda que contiene el dato almacenado en el nodo raíz del árbol, un puntero (izdo) a una celda que representa el subárbol izquierdo del árbol de partida y un puntero (dcho) a una celda que representa el subárbol derecho del árbol de partida. Si un arbolBin no tiene subárbol izquierdo o derecho entonces el puntero correspondiente apunta a NULL. Si un arbolBin es vacío el puntero apunta a NULL.

• IMPLEMENTACIÓN DE LAS OPERACIONES:

```
accion iniciarArbol( sal A : arbolBin )  
  { Inicia A como un árbol vacío }  
  principio  
    A ← NULL  
  fin
```

```
accion enraizar( sal A : arbolBin, E/S Aiz : arbolBin,  
                 E/S Ade : arbolBin, ent d : telemento )  
{ Construye el árbol A en cuya raíz se sitúa el dato d  
  del cual penden los árboles Aiz y Ade. El acceso a  
  los subárboles de A mediante Aiz y Ade queda anulado}
```

```
principio  
  A ← reservar(Celda)  
  si A ≠ NULL entonces  
    dest(A).dato ← d  
    dest(A).izdo ← Aiz  
    dest(A).dcho ← Ade  
    Aiz ← NULL  
    Ade ← NULL  
  fsi  
fin
```

```
funcion izquierdo( A : arbolBin ) devuelve arbolBin  
{Devuelve el árbol izquierdo que pende del nodo raíz de A}
```

```
principio  
  devuelve( dest(A).izdo )  
fin
```

```
funcion derecho( A : arbolBin ) devuelve arbolBin  
{Devuelve el árbol derecho que pende del nodo raíz de A }
```

```
principio  
  devuelve( dest(A).dcho )  
fin
```

```
funcion raiz( A : arbolBin ) devuelve telemento  
{ Devuelve el dato situado como nodo raíz del árbol A }
```

```
principio  
  devuelve( dest(A).dato )  
fin
```

```
funcion arbolVacio( A : arbolBin ) devuelve booleano  
{ Si A está vacío devuelve VERDAD y FALSO en otro caso}
```

```
principio  
  devuelve( A=NULL )  
fin
```