

Tema 4. Estructuras de datos no lineales

Algoritmo: borrado iterativo de un nodo en un árbol binario de búsqueda

```
procedimiento Borrar(ref arbol : a; valor e : tipoElemento)
var
  arbol : aux, act, ant
inicio
  //Buscar el elemento
  //El procedimiento Buscar, devuelve la posición del elemento
  //en el puntero act y la dirección del predecesor en el puntero ant
  Buscar(a,e,act,ant)

  // si borramos una hoja
  si (act↑.hder = nulo) y (act↑.hizq = nulo) entonces
    si ant = nulo entonces
      // Es el nodo raíz
      a ← nulo
    si_no
      si ant↑.hder = act entonces
        // es el subarbol derecho del anterior
        ant↑.hder ← nulo
      si_no
        // es el subarbol izquierdo del anterior
        ant↑.hizq ← nulo
      fin_si
    fin_si
  si_no
    // si el nodo a borrar tiene dos hijos
    si (act↑.hder <> nulo) y (act↑.hizq <> nulo) entonces
      // inicializa los punteros para buscar
      // el valor a reemplazar
      ant ← act
      aux ← act↑.hizq
      // localiza el nodo que contiene el valor
      // más próximo al que se va a suprimir
      mientras aux↑.hder <> nulo hacer
        ant ← aux
        aux ← aux↑.hder
      fin_mientras
      // pone el valor más próximo en el nodo
      // cuyo valor se va a suprimir
      act↑.raiz ← aux↑.raiz
      // se suprime el nodo del que se ha tomado el valor
      si ant = act entonces
        //El nodo a suprimir está justo a la izquierda del elemento a borrar
        ant↑.hizq ← aux↑.hizq
      si_no
        ant↑.hder ← aux↑.hizq
      fin_si
      act ← aux // para poder liberar el nodo
    si_no
      // solo tiene un hijo
      // se actualizan los punteros del nodo a borrar
      // según tenga un hijo izquierdo o un hijo derecho
      si act↑.hder <> nulo entonces
        // act tiene un hijo derecho
        si ant = nulo entonces
          // se trata del nodo raíz
          a ← act↑.hder
```

```

    si_no
      // se trata de un nodo no raíz
      si ant↑.hder = act entonces
        // se trata del hijo derecho del anterior
        ant↑.hder ← act↑.hder
      si_no
        // se trata del hijo izquierdo del anterior
        ant↑.hizq ← act↑.hder
      fin_si
    fin_si
  si_no
    // act tiene un hijo izquierdo
    si ant = nulo entonces
      // se trata de un nodo raíz
      a ← act↑.hizq
    si_no
      // se trata de un nodo no raíz
      si ant↑.hder = act entonces
        // se trata del hijo derecho del anterior
        ant↑.hder ← act↑.hizq
      si_no
        // se trata del hijo izquierdo del anterior
        ant↑.hizq ← act↑.hizq
      fin_si
    fin_si
  fin_si
  fin_si
  liberar(act)
fin_procedimiento

```