

Listas y árboles

Algoritmos – TUPAR

Estructuras dinámicas

- ▶ Si bien php tiene la posibilidad de definir arreglos dinámicos, es importante saber utilizar estructuras para manejar memoria dinámica.

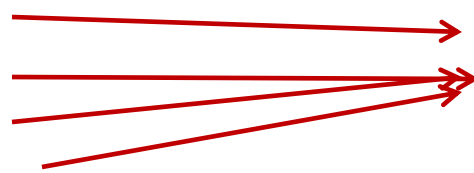
```
$arreglo=array();
```

```
$arreglo[]=2;
```

```
$arreglo[]=4;
```

```
$arreglo[]=3;
```

```
$arreglo[]=6;
```

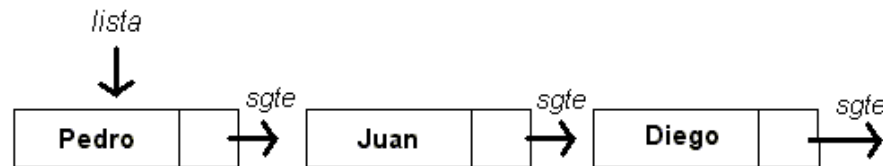


Inserta al final del
arreglo

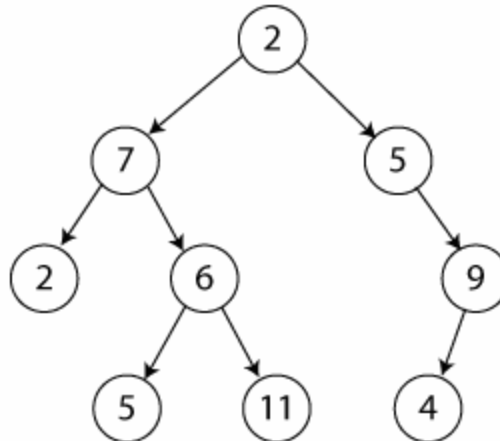
Estructuras dinámicas

- ▶ Otra posibilidad es usar estructuras para definir Listas vinculadas o Árboles para implementar determinados algoritmos

- Listas



- Árboles



Listas

- ▶ Definir estructura de datos que contenga información y un puntero a un elemento siguiente. Este será cada nodo de una lista:

```
class NodoLista{  
    public $dato;  
    public $sgte;  
}
```

Listas

- ▶ Definir un puntero a NodoLista.
Si está en NULL, significa que la lista está vacía.
Para agregar elementos, debería utilizar new en el puntero correspondiente:

```
$lista=null;
```

```
//..
```

```
$lista=new NodoLista;
```

```
$lista->dato=3;
```

```
$lista->sgte=null;
```

Listas

- ▶ Definir un puntero a NodoLista.
Si está en NULL, significa que la lista está vacía.

Para agregar elementos, debería utilizar new en el puntero correspondiente:

```
$lista=null;
```

```
//..
```

```
$lista=new NodoLista;  
$lista->dato=3;  
$lista->sgte=null;
```

miLista 

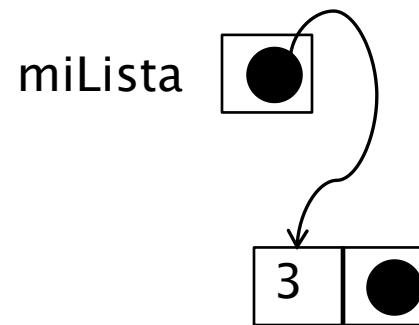
Listas

- ▶ Definir un puntero a NodoLista.
Si está en NULL, significa que la lista está vacía.
Para agregar elementos, debería utilizar new en el puntero correspondiente:

```
$lista=null;
```

```
//..
```

```
$lista=new NodoLista;  
$lista->dato=3;  
$lista->sgte=null;
```



Listas

- ▶ Para agregar un elemento a la lista, debo recorrer la lista, hasta encontrar el nodo que tiene el puntero 'sgte' en NULL, y crear un nodo a partir de él:

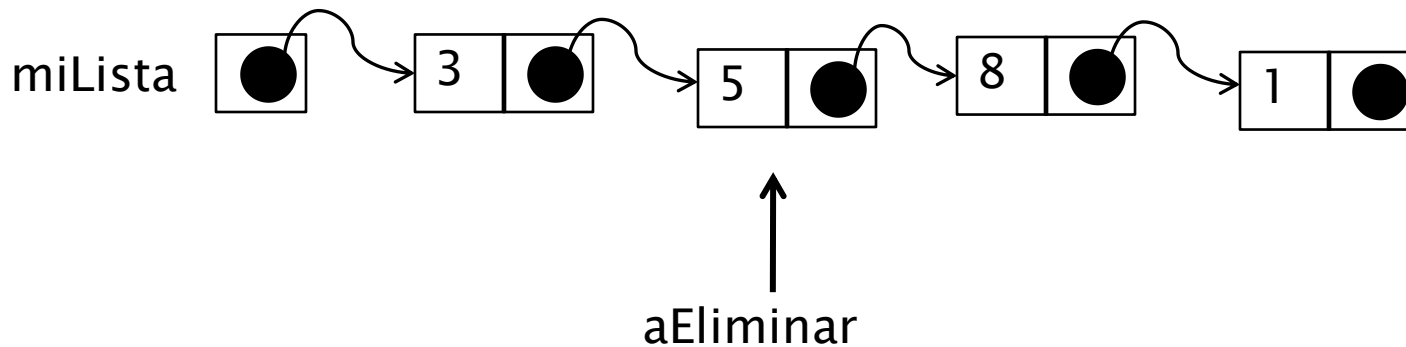
- Iterativo (usando cursor).

- Recursivo.

(Ver ejemplos)

Listas

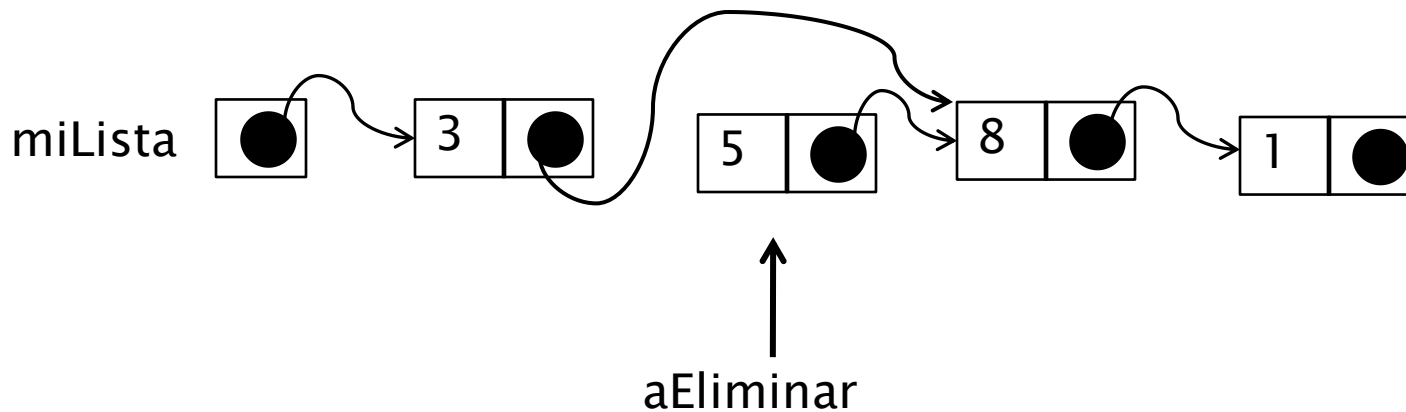
- ▶ Para eliminar un nodo:
 - Encontrar el nodo a eliminar.
 - Actualizar punteros.
 - Liberar memoria (no necesario en php).



Encuentro Nodo
a eliminar

Listas

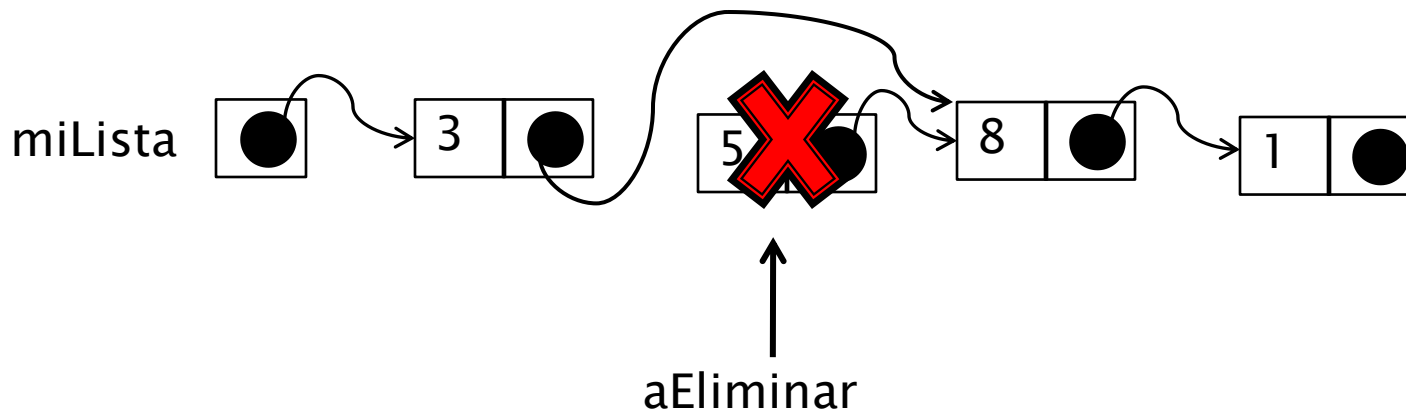
- ▶ Para eliminar un nodo:
 - Encontrar el nodo a eliminar.
 - Actualizar punteros.
 - Liberar memoria (no necesario en php).



Actualizo punteros

Listas

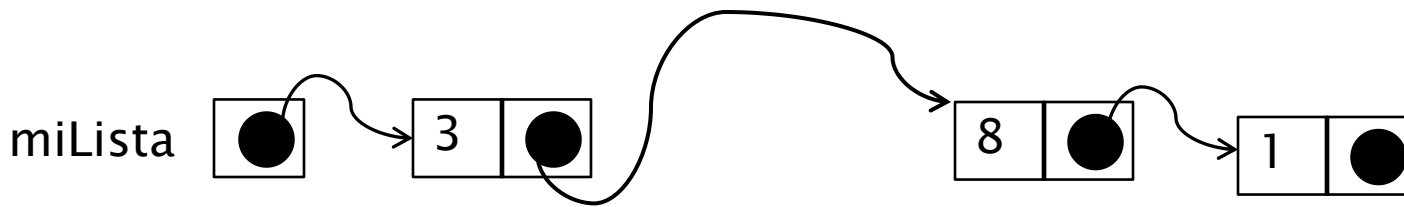
- ▶ Para eliminar un nodo:
 - Encontrar el nodo a eliminar.
 - Actualizar punteros.
 - Liberar memoria (no necesario en php).



Elimino el nodo

Listas

- ▶ Para eliminar un nodo:
 - Encontrar el nodo a eliminar.
 - Actualizar punteros.
 - Liberar memoria (no necesario en php).



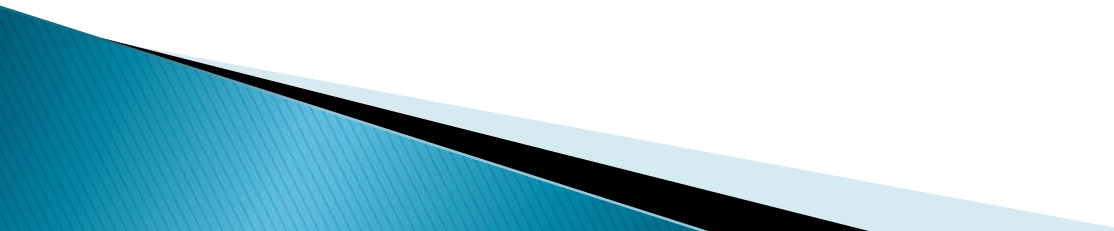
Elimino el nodo

Arboles

- ▶ La estructura de un árbol (binario) involucra un nodo que guarda un dato y un puntero al sub-árbol izquierdo y otro al sub-árbol derecho:

```
class NodoArbol{  
    public $dato;  
    public $der;  
    public $izq;  
}
```

Arboles

- ▶ Se define un puntero a `NodoArbol`. Si está en `NULL`, el árbol está vacío.
 - ▶ Los nodos se insertan como hojas.
 - ▶ Los árboles son estructuras recursivas. Todo recorrido en él, requiere un algoritmo recursivo (ver códigos de ejemplo).
- 

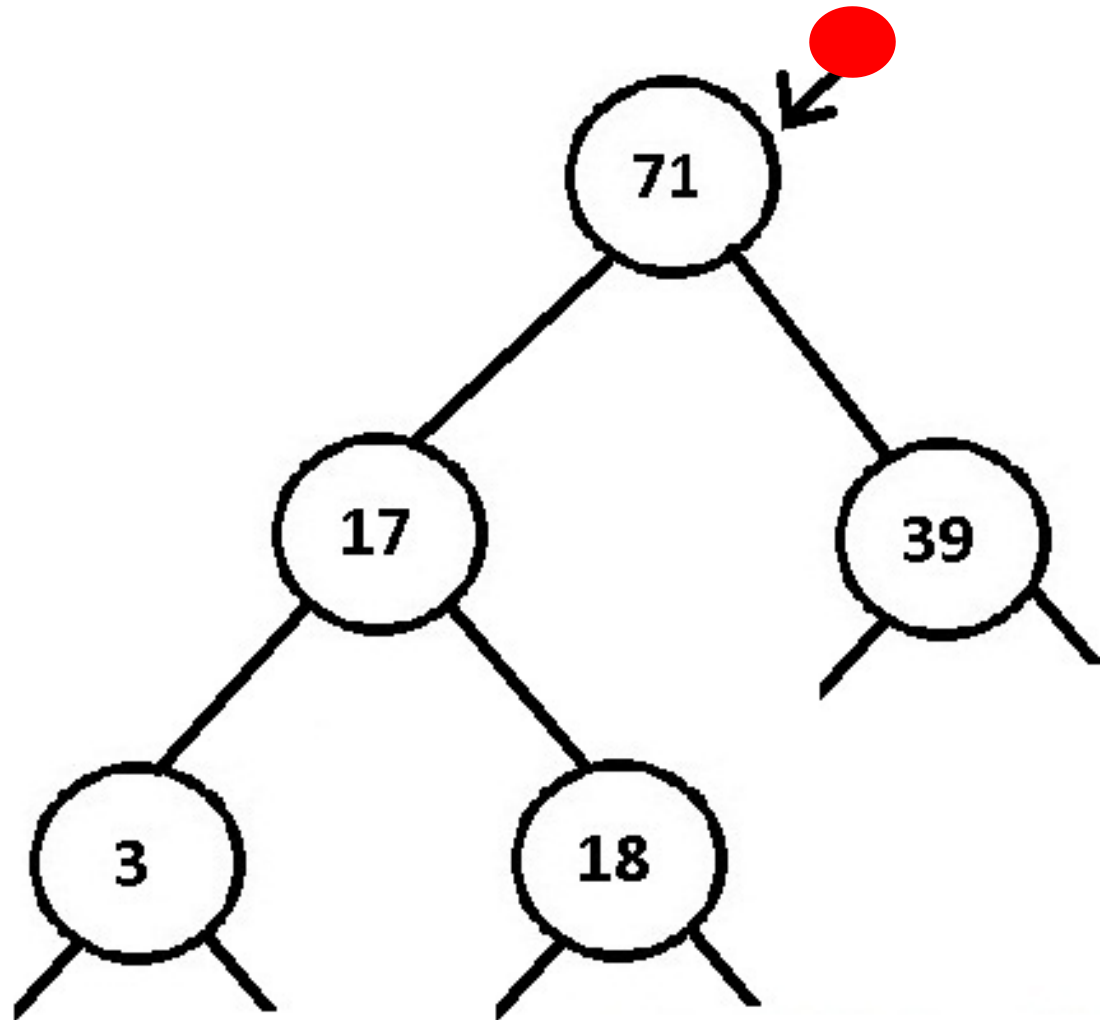
Arboles

- ▶ Insertar ordenado:

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```

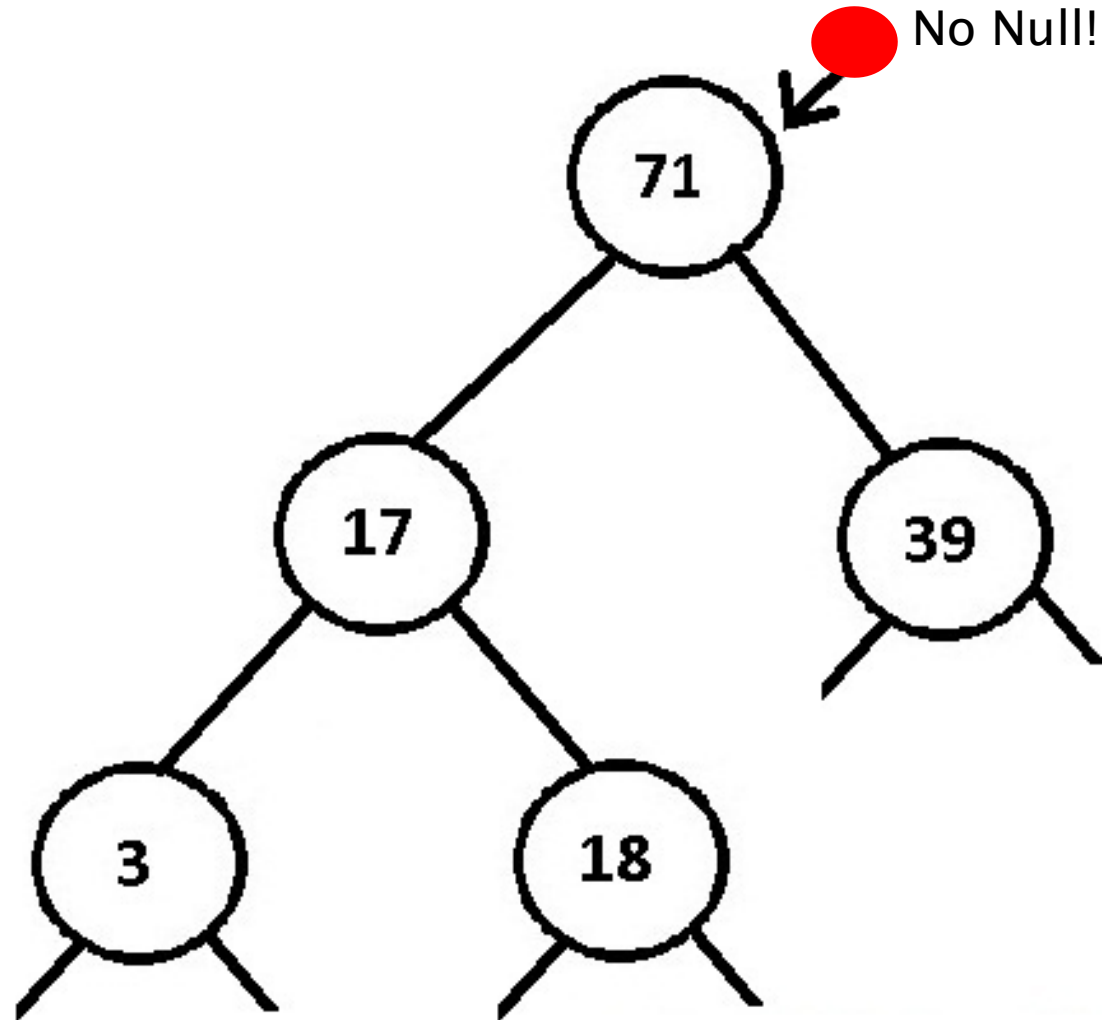
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



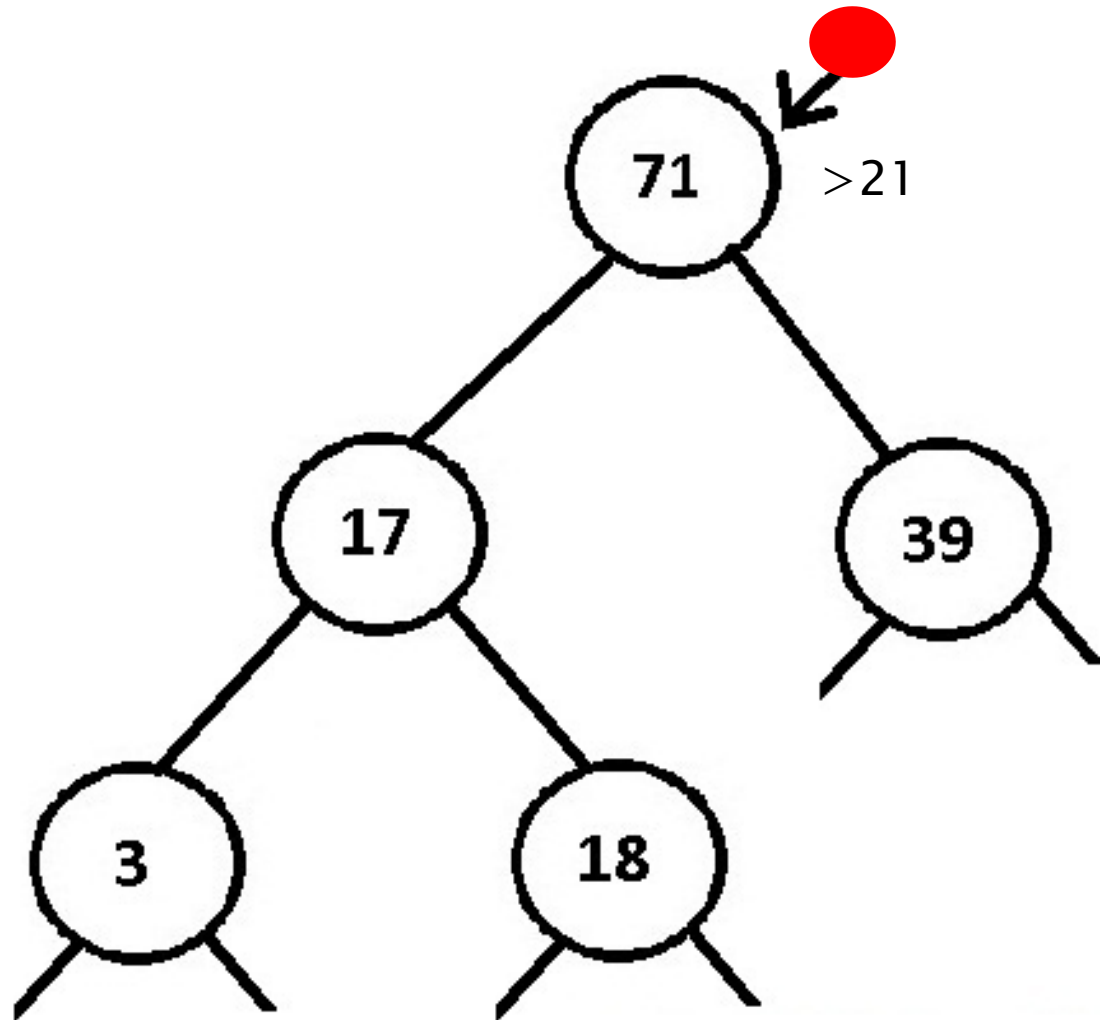
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



Arboles

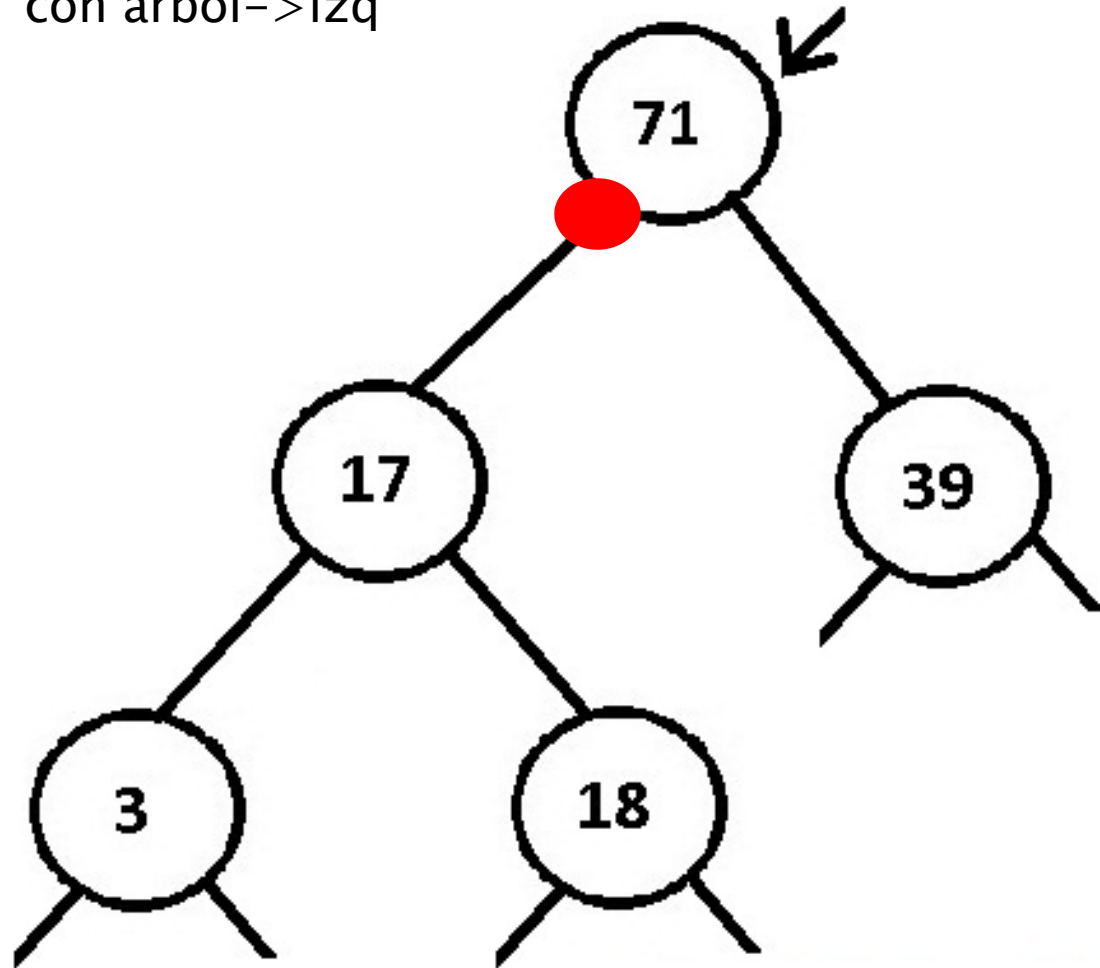
```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



Arboles

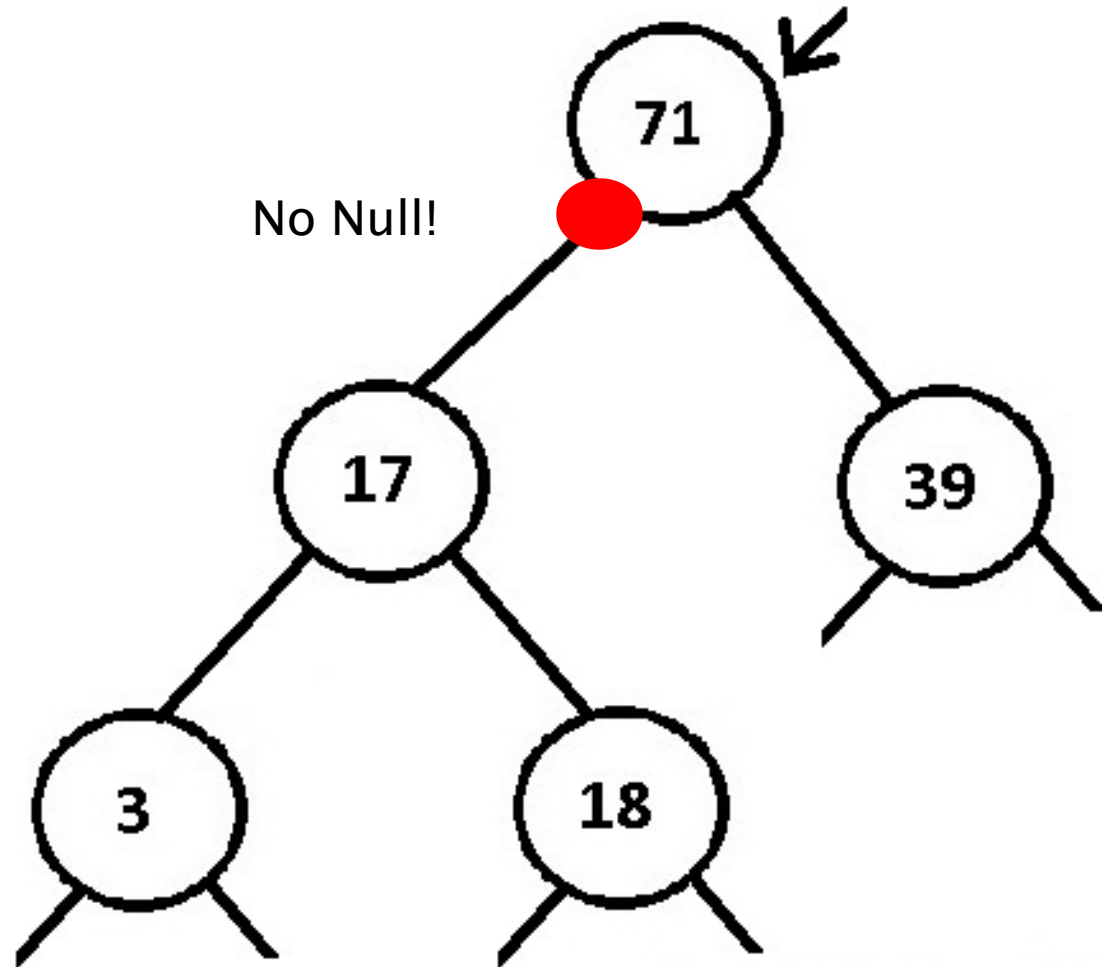
Llamo recursivo
con `arbol->izq`

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



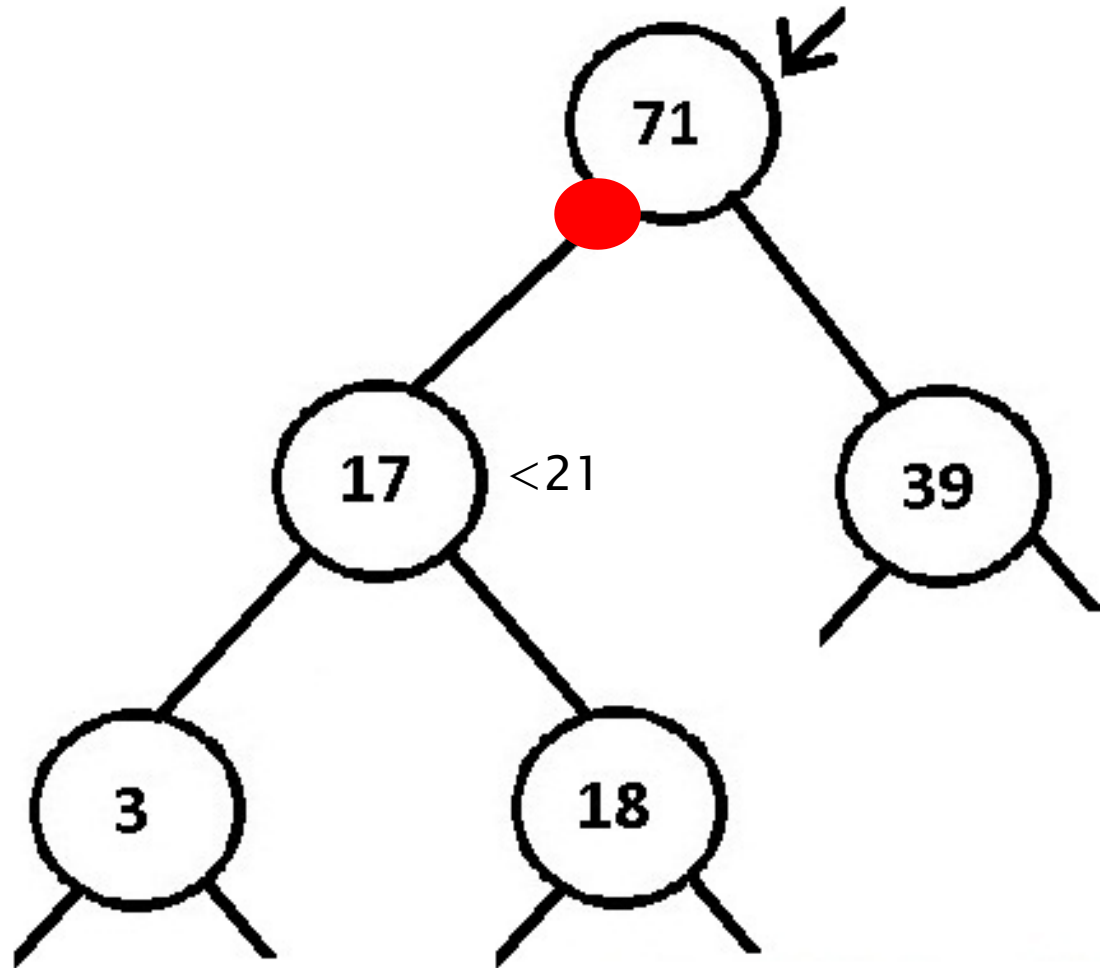
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



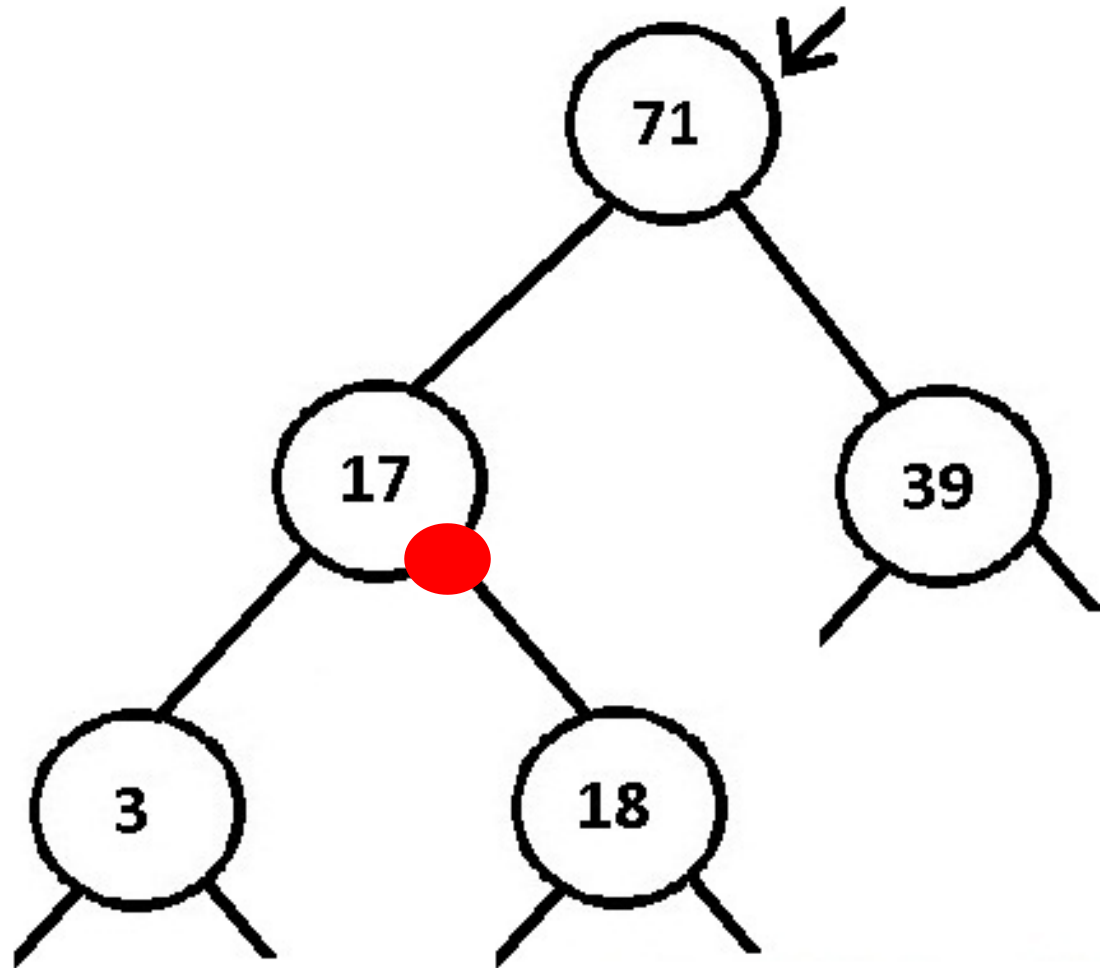
Arboles

```
function addNodo(&$arb, $dato){  
  if ($arb==null){  
    $arb=new NodoArbol;  
    $arb->dato=$dato;  
    $arb->izq=null;  
    $arb->der=null;  
  }  
  else  
    if ($arb->dato<$dato)  
      addNodo($arb->der, $dato);  
    else  
      addNodo($arb->izq, $dato);  
}
```



Arboles

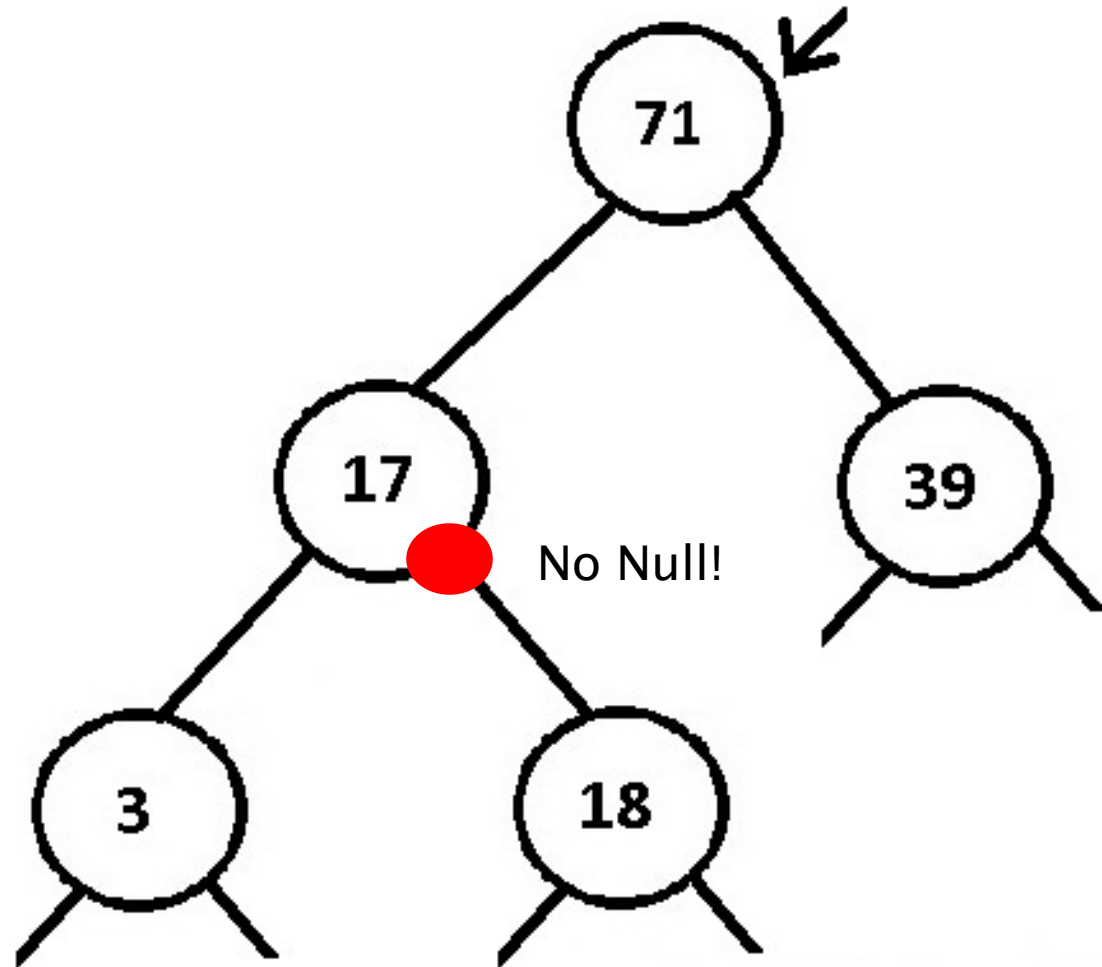
```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



Llamado recursivo

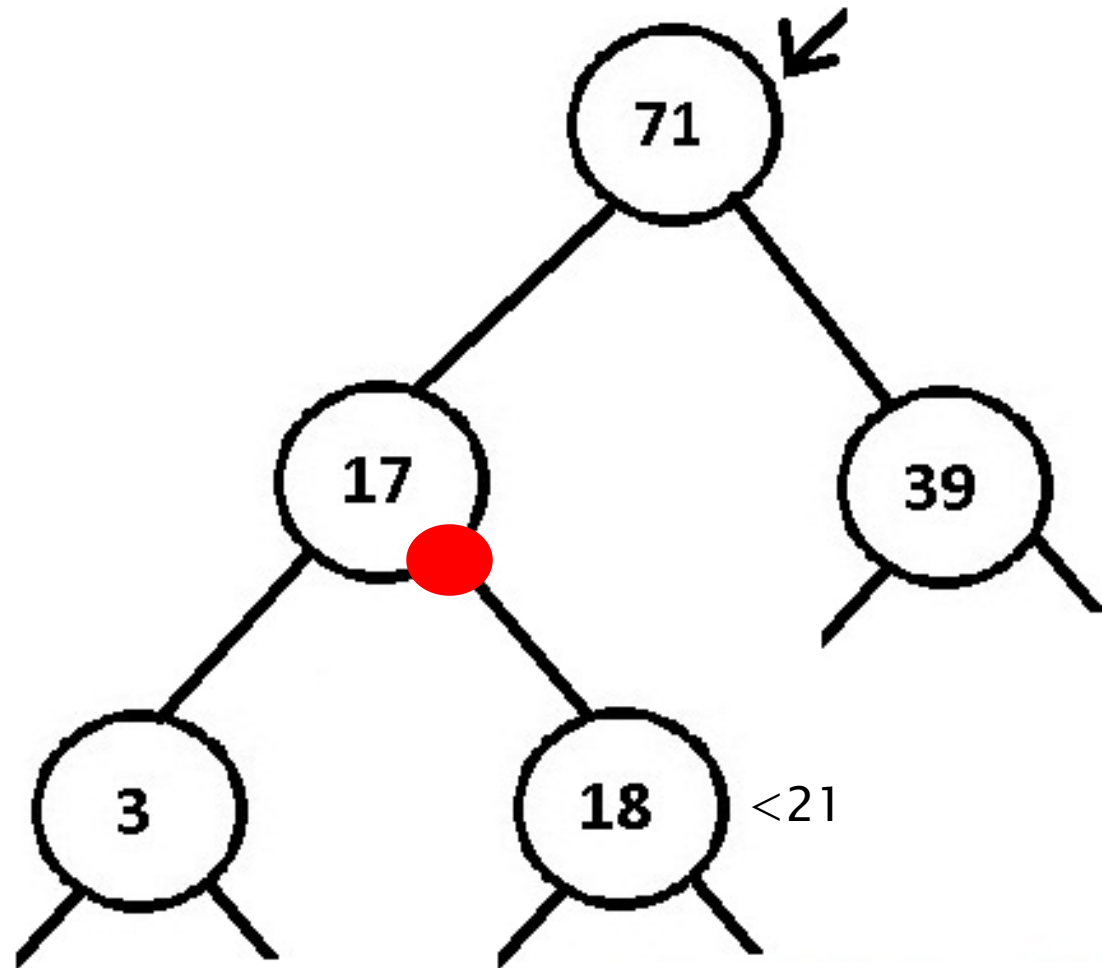
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



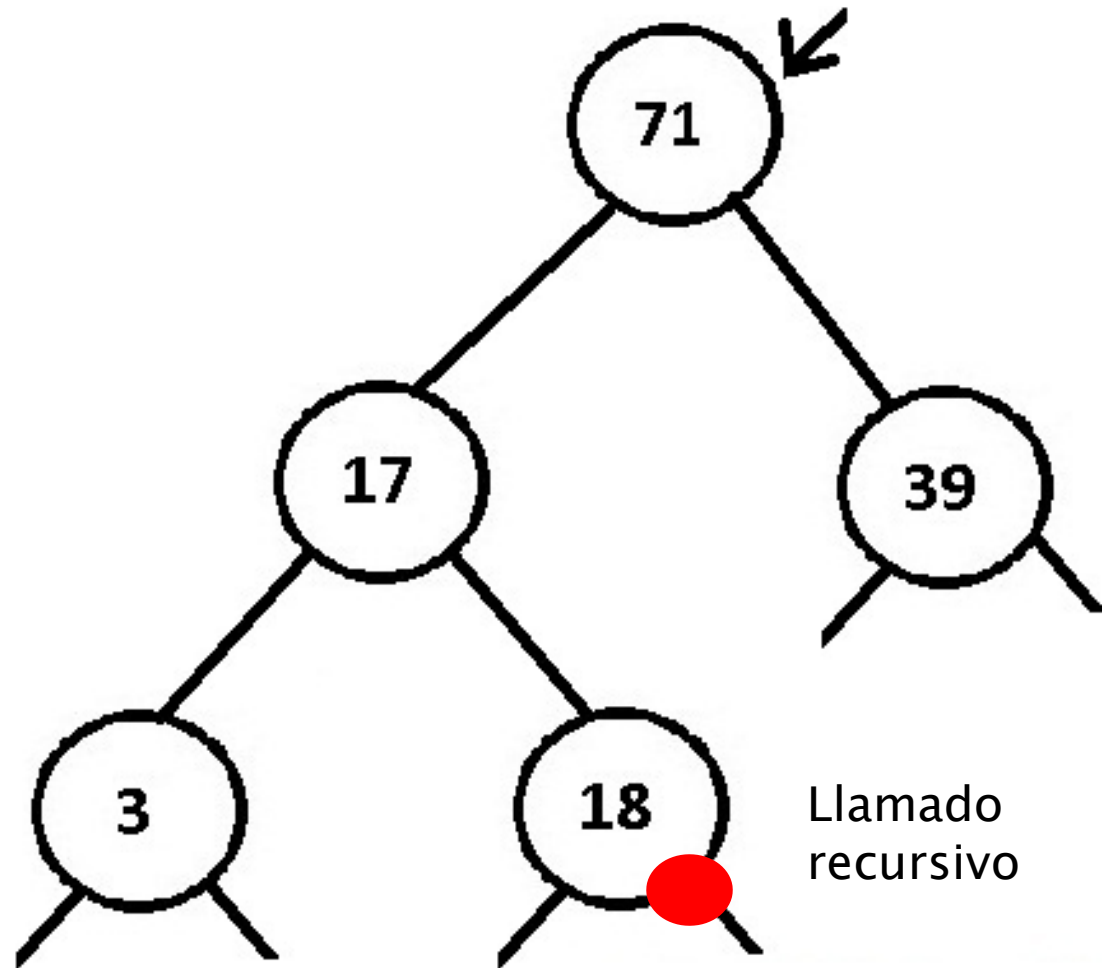
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



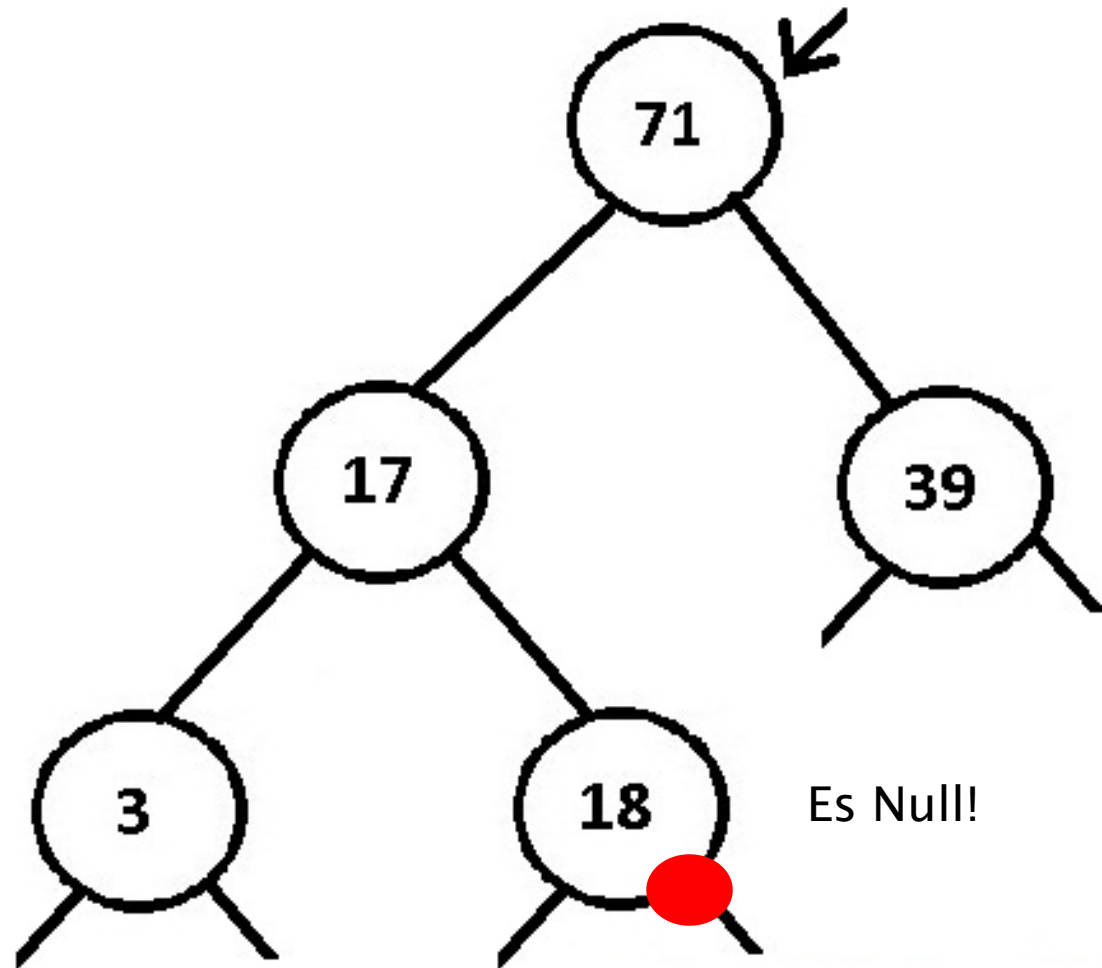
Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



Arboles

```
function addNodo(&$arb, $dato){  
    if ($arb==null){  
        $arb=new NodoArbol;  
        $arb->dato=$dato;  
        $arb->izq=null;  
        $arb->der=null;  
    }  
    else  
        if ($arb->dato<$dato)  
            addNodo($arb->der, $dato);  
        else  
            addNodo($arb->izq, $dato);  
}
```



Arboles

```
function addNodo(&$arb, $dato){  
  if ($arb==null){  
    $arb=new NodoArbol;  
    $arb->dato=$dato;  
    $arb->izq=null;  
    $arb->der=null;  
  }  
  else  
    if ($arb->dato<$dato)  
      addNodo($arb->der, $dato);  
    else  
      addNodo($arb->izq, $dato);  
}
```

