

Video22 (1)

Conceptos Principales

1. a
2. b
3. c

Notas

Algoritmo de búsqueda en Árbol Binario de Búsqueda (ABB)

- La búsqueda en un **ABB** sigue la misma lógica de su construcción:
 - **Si el valor buscado < valor del nodo actual → ir a la izquierda.**
 - **Si el valor buscado > valor del nodo actual → ir a la derecha.**
 - **Si el valor buscado = valor del nodo actual → encontrado.**
 - **Si llegamos a un nodo nulo → no está en el árbol.**

Ejemplo práctico

- Buscar **70** en un ABB:
 1. Raíz = 48 → 70 es mayor → ir derecha (83).
 2. Nodo 83 → 70 es menor → ir izquierda (75).
 3. Nodo 75 → 70 es menor → ir izquierda (51).
 4. Nodo 51 → 70 es mayor → ir derecha (70).
 5. Nodo 70 → encontrado.
- Buscar **45** en el mismo árbol:
 1. Raíz = 48 → 45 es menor → izquierda (36).
 2. Nodo 36 → 45 es mayor → derecha (31).
 3. Nodo 31 → 45 es mayor → derecha (nulo).
 4. Llegamos a nulo → **no existe en el árbol**

Pseudocódigo (recursivo)

```
func buscar(nodo, valor):
    si nodo == NULL:
        return NULL // no encontrado
    si valor == nodo.valor:
        return nodo // encontrado
    si valor < nodo.valor:
        return buscar(nodo.izquierdo, valor)
    sino:
```

```
return buscar(nodo.derecho, valor)
```

- Retorna el **puntero al nodo** si lo encuentra, o **NULL** si no existe.
- Importante: en recursividad siempre usar **return** en las llamadas, de lo contrario se pierde el resultado.