

### Reflexión Act 3.4

Definitivamente, la implementación de un *binary search tree* tiene muchas bondades, entre las que se encuentra que, se maneja con memoria dinámica, es decir, no tiene un tamaño preestablecido y puede ajustar su tamaño en memoria gracias al *heap*.

Adicionalmente, otra de sus ventajas se encuentra en las distintas formas de recorrer los nodos y procesarlos. Dependiendo de la situación, se puede recorrer en *preorder*, *postorder* o *inorder*, esto podría permitir identificar ciertos nodos que cuentan con una característica que se desea buscar en todo el recorrido.

Por último, en cuanto a su complejidad, en un caso promedio, los métodos de insertar, eliminar y buscar tienen una complejidad de  $O(\log n)$  cuando  $n$  es el número de nodos en el árbol. Sin embargo, es importante remarcar que, en el peor de los casos, se tiene una complejidad de  $O(n)$ , y esto pasa cuando el árbol está desbalanceado.