¿Por qué es importante que <a> tenga comparaciones?:  
El TAD ABB funciona bajo esta regla: "Para todo nodo, los valores del subárbol izquierdo son menores, y los del derecho son mayores al valor del nodo."  
Por lo tanto, las operaciones <, > y == son esenciales para:

* Insertar valores en la posición correcta.
* Buscar eficientemente.
* Eliminar nodos sin romper el orden.

Si el tipo <a> no admite comparaciones, entonces:

* No se puede mantener el invariante del árbol de búsqueda.
* No se puede decidir si un nuevo valor va a la izquierda o a la derecha.
* La estructura dejaría de ser un ABB, y pasaría a ser simplemente un árbol binario sin orden.

¿Qué operaciones modificaría o eliminaría?:



¿Qué operaciones podría agregar o reemplazar?:

* comparador\_externo(función): Permite pasar una función de comparación como parámetro genérico. Esto abstrae la lógica de comparación para que funcione con cualquier tipo.
* insertarIzquierda – insertarDerecha: Si no hay orden, podrías permitir que el usuario indique explícitamente dónde insertar.
* igual(a, b): Si no existe ==, podría definirse una función de igualdad personalizada.

¿Cómo afectaría esto a la implementación?:

* La lógica de inserción, búsqueda y eliminación debería cambiar completamente:
* Ya no se puede tomar decisiones automáticas.
* Se debe recorrer todo el árbol para buscar un valor.
* El árbol ya no funcionaría con búsqueda binaria eficiente O(log n):
* Las búsquedas y operaciones serían en el peor caso O(n).
* Se pierde la capacidad de balancear el árbol, porque no hay forma de saber el "orden relativo" entre elementos.

Conclusión:  
Si el tipo <a> no admite comparaciones, el árbol binario ya no puede actuar como árbol de búsqueda.

* Algunas operaciones deben modificarse drásticamente o eliminarse.
* Se podría transformarlo en un árbol general binario sin orden, o bien exigir que el tipo <a> implemente una función de comparación externa.