

Se ha proporcionado todo lo relativo a la implementación del BSTree, por lo que se pide que se EXPLIQUE en un documento (memoria) toda implementación suministrada, prestando especial atención a funciones principales de dicha estructura de datos como: *add*, *remove*, *clear*, *contains*, *isEmpty*, y al iterador *iterator* del tipo *Treeliterator*. Razone y justifique adecuadamente cada una de las ventajas e inconvenientes enumerados.

Add: Dado un elemento pasado por parámetro descendiendo desde la raíz buscando la posición donde debe encajarse dicho elemento. Para ello se comprueba si el elemento es mayor o menor (en caso de ser igual devuelve un false) que el valor del nodo donde se encuentra el iterador, si es mayor se desplaza al nodo hijo derecho y es menor se desplaza al nodo hijo izquierdo, si dichos nodos son nulos se agrega dicho elemento si no se repite dicha operación de desplazamiento mientras el nodo no sea nulo.

Clear: Vacía el árbol poniendo el contador a 0 y el nodo root apunta a nulo.

Remove: Se busca en el árbol si el elemento a eliminar esta y en caso afirmativo se llama al método `removeNode` y en caso negativo devuelve un false. En el método `removeNode` se comprueba si tiene hijos, si no los tiene se elimina pero en caso de tenerlos se debe comprobar si solo tiene uno, que en este caso dicho hijo ocupara el lugar en el árbol del nodo a eliminar, o si tiene dos hijos, en cuyo caso se debe comprobar si el hijo derecho tiene hijo izquierdo. Si el hijo derecho no tiene hijo izquierdo, el nodo hijo izquierdo del nodo a eliminar pasaría a ser el hijo izquierdo de su hermano y este el último el padre. En caso de que el hijo derecho del nodo a eliminar tengo hijo izquierdo, el hijo izquierdo del hijo derecho del raíz pasara a ser hijo derecho del hijo izquierdo del raíz y el hijo izquierdo del nodo raíz pasara a ser el hijo izquierdo del hijo derecho del raíz.

Contains: recorre el árbol y devuelve un true si está el elemento.

IsEmpty: Comprueba si el árbol está vacío viendo a donde apunta el nodo root.

Iterator: Devuelve un iterador que puede recorrer el arbol.

Treeliterator: implementa la interfaz iterator para reconocer el arbol.