**OBSERVACIONES DEL LA PRACTICA**

Estudiante 1 Cod 202421132

Estudiante 2 Cod 202414673

Estudiante 3 Cod 202422837

# **Preguntas de análisis**

1. ¿Qué diferencia existe entre las alturas de los dos árboles (BST y RBT)?

Un árbol BST puede llegar a tener una altura muy grande si los datos sin insertados de manera ascendente o descendente lo que puede llegar a hacerlo deziquiibrado, a diferencia del RBT, el cual se balancea durante las inserciones y elimaciones creando asi un árbol siempre balanceado.

1. ¿Percibe alguna diferencia entre la ejecución de los dos árboles (RBT y BST)? ¿Por qué pasa esto?

Las funciones de ordenamiento en un árbol RBT son mas rápidas y organizadas pues se balancea lo cual evita que las operaciones tengan que recorrer ramas demasiado largas como podría pasar en un BST desbalanceado.

1. ¿Existe alguna diferencia de complejidad entre los dos árboles (RBT y BST)? Justifique su respuesta.

Sí. Aunque ambos tienen complejidad O(log n) en promedio para operaciones de búsqueda, inserción y eliminación, en el peor caso un BST puede degradarse a una lista enlazada con complejidad O(n), mientras que un RBT mantiene siempre O(log n) gracias a su estructura balanceada. Esto hace que el RBT tenga mejor comportamiento.

1. ¿Existe alguna manera de cargar los datos en un árbol RBT de tal forma que su funcionamiento mejore? Si es así, mencione cuál.

Sí, para mejorar el rendimiento de un RBT al cargar datos, se recomienda evitar insertar los datos en orden ascendente o descendente. Una buena estrategia es insertar los datos en un orden aleatorio o utilizar un algoritmo de carga balanceada que simule un árbol perfectamente balanceado antes de insertar en el RBT, así se reducen al mínimo las rotaciones y recoloreos necesarios.