



# Tecnológico de Monterrey

**Programación de estructuras de datos y algoritmos fundamentales (Gpo 602)**

**Act 3.4 - Actividad Integral de BST (Evidencia Competencia)**

**investigación y reflexión de la importancia y eficiencia del uso de BST en una situación problema  
de esta naturaleza**

**Fecha**

28/10/2025

**Profesor**

Daniel Perez R

**Alumno**

Santiago Amir Rodriguez Gonzales A01739942

## **Investigación: árboles binarios de búsqueda**

Cuando se trata de manejar grandes volúmenes de información, los árboles binarios de búsqueda (BST) son como organizadores silenciosos que hacen que todo tenga sentido. Su estructura jerárquica permite que cada dato se ubique en el lugar correcto: si es menor, va a la izquierda; si es mayor, a la derecha. Esta lógica simple permite realizar búsquedas, inserciones y eliminaciones de forma muy eficiente, incluso cuando los datos crecen sin parar.

Una de sus grandes ventajas es que mantienen el orden de los datos sin necesidad de aplicar algoritmos externos. Esto es clave en sistemas que están en constante actualización, como las bitácoras de accesos o los registros de red. Además, si los datos tienen varios criterios el BST puede adaptarse para organizarlos jerárquicamente, aplicando reglas de desempate que permiten mantener la coherencia incluso cuando hay valores repetidos.

Comparado con estructuras lineales (listas o arreglos), el BST ofrece una mejora significativa en rendimiento. Mientras que en una lista la búsqueda puede tardar proporcionalmente al número de elementos ( $O(n)$ ), en un BST bien balanceado ese tiempo se reduce a  $O(\log n)$ . Esto lo convierte en una herramienta ideal para sistemas que requieren rapidez y precisión, como bases de datos, análisis de tráfico en redes y sistemas de ciberseguridad.

Continuando con el tema de seguridad, los BST también pueden ser aliados en la detección de infecciones en una red. ¿Cómo? Al organizar los registros de acceso de forma ordenada, es posible identificar patrones inusuales: por ejemplo, una IP que aparece repetidamente en horarios extraños, o accesos que se desvían del comportamiento habitual. Si los datos estuvieran desordenados, esos detalles podrían pasar desapercibidos. Pero con un BST, se pueden recorrer los registros en orden, aplicar filtros y detectar anomalías que podrían indicar que la red está comprometida.

En resumen, los árboles binarios de búsqueda no solo almacenan datos: los organizan, los hacen accesibles y permiten analizarlos de forma inteligente. Son una herramienta poderosa para cualquier sistema que necesite rapidez, orden y capacidad de escalar sin perder eficiencia.

## **Reflexión**

Trabajar con un árbol binario de búsqueda me hizo ver que elegir una estructura de datos no es solo una decisión técnica: es una forma de pensar en cómo resolver problemas reales. En el proyecto, que consistía en analizar una bitácora de accesos, entendí que no basta con guardar los datos. Hay que organizarlos de manera que se puedan recorrer, comparar y entender. Ahí es donde el BST se volvió esencial.

Lo que más me sorprendió fue cómo el árbol no solo ordena la información, sino que la presenta de forma lógica. Si los datos estuvieran desordenados, muchos detalles pasarían desapercibidos. Me di cuenta de que las estructuras de datos también pueden ser herramientas de seguridad, ayudando a identificar posibles infecciones o accesos no autorizados en una red.

Entonces, una forma efectiva de detectar infecciones de red es analizar los registros de acceso con ayuda de un BST. Al recorrer los datos en orden, puedes identificar patrones que no encajan. Y lo mejor es que el BST permite hacer este análisis de forma rápida y eficiente, incluso cuando hay miles o millones de registros.

Otro aprendizaje importante fue sobre eficiencia y escalabilidad. Un programa puede funcionar bien con pocos datos, pero si la estructura no está bien pensada, se vuelve lento y poco funcional cuando el volumen crece. Este proyecto me enseñó que no basta con que el código funcione: tiene que hacerlo bien, incluso cuando el problema se vuelve grande. Elegir una estructura adecuada es como construir los cimientos de una casa: si están mal hechos, todo lo demás se tambalea.

Finalmente, esta experiencia me ayudó a ver la programación desde una perspectiva más analítica. Ya no se trata solo de escribir código, sino de entender cómo cada decisión impacta en la solución. Un árbol binario, que al principio parece algo simple, puede marcar la diferencia entre un sistema que solo guarda datos y uno que realmente los entiende, los analiza y ayuda a tomar decisiones importantes.