



## **Act 3.4: Reflexión sobre Binary Search Trees**

*Instituto Tecnológico y de Estudios Superiores de Monterrey*

Programación de estructuras de datos y algoritmos fundamentales

22/10/2021

**Profesores:**

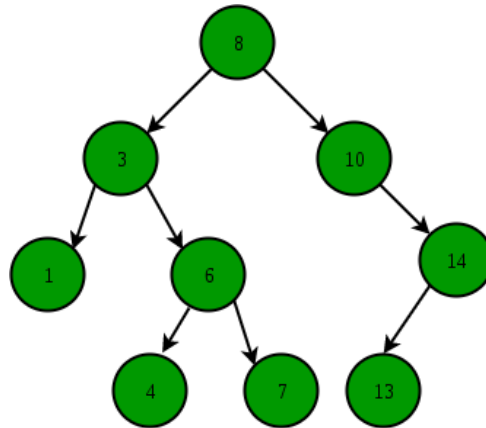
Luis Ricardo Peña Llamas

**Elaborado por:**

Pablo Agustín Ortega Kral A00344664

## Reflexión sobre Binary Search Trees

Un árbol binario de búsqueda es una estructura de datos cuya unidad fundamental es nodo que consiste de un dato, dos apuntadores hacia sus nodos hijos (izquierda y derecha) y un valor llamado *key* que hace referencia al peso o prioridad del valor. De esta forma, los BST (*Binary Search Trees*) se forman a partir de un nodo raíz y colocan los siguientes valores dependiendo del *key* que se le asignen: poniendo los de mayor *key* en el subárbol derecho y los de menor en el izquierdo.



*Imagen 1. Ejemplo de un BST*

La primera ventaja aparente de estos árboles es que nos permite almacenar la información de manera ordenada, pudiendo acceder a los elementos de manera ascendente haciendo un recorrido en *preorden*. En el aspecto de la complejidad temporal, la utilidad de tener la información organizada en dicha estructura de datos recae en que las operaciones de inserción, borrado y búsqueda ahora dependen de la altura del árbol en lugar del número de elementos que contienen; cuando se implementan algoritmos para optimizar la altura de los árboles binarios, es decir balancearlos para que se trate de un árbol completo resultando en una complejidad de  $O(\log N)$ .

En el proyecto, implementamos los árboles binarios de búsqueda para organizar las IP por números de accesos y después recorrimos el árbol de tal forma que los desplegara en orden ascendente para conocer los equipos que realizaban el mayor número de accesos. Relacionando esto a la detección de Ataques DDOS, que buscan saturar un equipo haciendo múltiples *requests*, podríamos revisar las IP con mayores accesos para identificar posibles equipos que forman parte de una botnet y como tal tienen un número elevado de peticiones al servidor.

**Referencias:**

- Applications of BST - GeeksforGeeks. (2021). Retrieved 22 October 2021, from <https://www.geeksforgeeks.org/applications-of-bst/>