



**Programación de estructuras de datos y
algoritmos fundamentales (Gpo 14)**

Act 3.4 - Actividad Integral de BST (Evidencia Competencia)

Alumno:

Alejandro Daniel González Carrillo	A01570396
Agustín Blanco Oliver	A00828415

Profesores:

Luis Humberto González Guerra

© 2020 Derechos reservados: Ninguna parte de esta obra puede ser reproducida o transmitida, mediante ningún sistema o método, electrónico o mecánico, sin conocimiento por escrito de los autores.

Monterrey, Nuevo León. 25 de octubre de 2020.

Reflexion: Para este proyecto se nos requirio utilizar busqueda de arboles binarios para lograr encontrar los 5 ips mas accesadas de una bitacora donde las ips ya estaban ordenas. Durante el desarrollo, lo que hicimos fue leer los datos de la bitacora ya ordenada, y mandar los datos al arbol donde serian ordenados de manera inorden converso de acuerdo a su cantidad de acceso que tuvieran. La importancia de utilizar BST en estos casos es que nos permite organizar las informacion de manera jerarquica, de la misma manera nos permite tener un manejo de memoria eficiente, ya que no reserva más memoria de la que necesita. Lo eficiente de usar BST es que podemos lograr una complejidad de $O(\log(n))$. Como podriamos determinar si una red esta infectada o no? Considerando la cantidad de veces que se intento acceder podemos saber que esa ip esta siendo mas atacada. Para lograr nuestra entrega en el main realizamos 3 funciones el leer datos que guarda toda la informacion de la bitacora en el programa. El ipToLong transforma el formato ip a un ip sin puerto y puntos de division. El ipToString transforma el formato ip y le remueve el puerto al ip. El BST.h tiene 5 metodos el constructor, destructor, add, inOrden y printInorden. El add añade un nodo al arbol con la ip (sin puerto), ip (valor long) y cantidad de accesos. El inOrden organiza de manera inorden converso los datos y imprime los 5 ips mas accesadas.

```
PS C:\Users\AlexGzz\Desktop\TC1031-Proyecto_Final-Act3.4> & .\"main.exe"
1: 10.15.176.241 38
2: 10.15.187.246 38
3: 10.15.176.230 37
4: 10.15.177.224 37
5: 10.15.183.241 37
PS C:\Users\AlexGzz\Desktop\TC1031-Proyecto_Final-Act3.4> |
```

Referencias:

Iteramos. (2010). Ventajas de los árboles de búsqueda binaria sobre tablas Hash. Recuperado el 25 de octubre del 2020 en <https://www.iteramos.com/pregunta/39322/ventajas-de-los-arboles-de-busqueda-binaria-sobre-tablas-hash>