



**Tecnológico  
de Monterrey**

Instituto Tecnológico y de Estudios Superiores de Monterrey  
Campus Monterrey

TC1031 Gpo. 606 - Programacion de estructuras de datos y algoritmos fundamentales

Profesora: Dra. Maria Valentina Narvaez Teran

Alumno: Adrian Hernandez Peña

### **Investigación:**

Las listas enlazadas tienen diferentes aplicaciones en diferentes softwares. Pueden servir desde una alternativa para organizar información como imágenes y canciones hasta una estructura de datos vital para la implementación de otras mucho más complejas como una *Queue*, *Stack*, o un *ordered set*. Las capacidades que estas estructuras de datos dan son fundamentales para algoritmos más avanzados como el dfs de grafos entre otros.

### **Reflexión:**

El problema con el que nos encontramos es la organización de una base de datos de eventos que son potenciales ataques a un servidor con respecto a fechas o ips. Es relevante la organización de esta información porque puede permitir detectar vulnerabilidades y la fecha exacta en que alguien pudo haberse infiltrado en el servidor para mitigar los daños y saber que medidas tomar posteriormente. Resulta vital tener un algoritmo que logre esto eficientemente porque permitiera trabajar con mayor rapidez a diferentes personas que requieran de la información.

Para resolver este problema se crearon dos clases diferentes, una representativa de un evento y otra representativa de un registro de eventos y encargada de organizarla. Primero se obtiene la ip como un arreglo de información y como un solo número representativo de la ip donde cada segmento está multiplicado por mil elevado a su posición desde la derecha hacia la izquierda, esto es para que todas las ips mapeadas a un solo número mantengan el mismo orden relativo entre ellas. Posteriormente se ordenan los eventos usando un algoritmo inspirado en radix sort en el cual se toma cada segmento de los cuatro diferentes de una IP como un solo dígito. Debido a que nuestros eventos son diferentes entre sí, para implementar las buckets del Radix sort implementamos nuestra pila que nos permite insertar y obtener información en tiempo constante. El algoritmo se repite en los segmentos que haya identificado que comparten el mismo dígito representativo según el que se está ordenando y después se ordenan dichos segmentos tomando en cuenta un dígito representativo a la derecha si es que existe uno.

Una vez ordenados los eventos pedimos dos ips que zanján el rango en el que daremos los eventos. Encontramos la ip menor de las dos probadas con búsqueda binaria en tiempo  $O(\log n)$  y después proveemos todos los datos que quepan en el rango de las ips linealmente después de ellas.

Las posibles mejoras que se pueden realizar en el código es una refactorización para usar funciones como argumento en una sola búsqueda binaria y no dos separadas, una para la fecha y otra para la ip. Una mejor interfaz en el menú puede también ser programada.

## Referencias:

- Applications of Linked List Data Structure | Linked List | Prepbytes. (2021, October 8). PrepBytes Blog.  
<https://www.prepbytes.com/blog/linked-list/applications-of-linked-list-data-structure/>