



**Programación de estructura de datos y algoritmos fundamentales**

Actividad 2.3 Actividad Integral estructura de datos lineales

Sebastian Antonio Almanza A01749694

Tecnológico de Monterrey

25 de Enero del 2024

Grupo 570

Profesor: Eduardo Arturo Rodriguez Tello

Hoy en día el buscar un contacto en nuestro celular, realizar una búsqueda dentro de los archivos de la computadora y realizar una búsqueda en un navegador son actividades que hoy consideramos cotidianas en nuestro constante uso de la tecnología, estas actividades tienen detrás el uso de estructuras de datos. El uso eficiente de estructuras de datos lineales nos permiten que el realizar búsquedas de información sea un proceso casi instantáneo, esto considerando que hemos elegido los métodos de ordenamiento y búsqueda más eficientes para que estos procesos se realicen de manera ágil. Para nuestra actividad integradora se nos dio una bitácora de registros de conexión y se nos pidió ordenarla y realizar búsquedas sobre ella. Para la implementación de la estructura de datos se tomaron en cuenta las siguientes:

- **Lista ligada:**

Una lista ligada es una estructura de datos lineal en la cual sus elementos no están guardados en una locación continua sino que están enlazados mediante apuntadores. En una lista ligada se forma una serie de nodos conectados en donde cada nodo contiene la información y el apuntador al siguiente nodo. (*“Understanding the basics of Linked list”*, 2024).

- **Lista doblemente ligada:**

Una lista doblemente ligada es de igual forma una estructura de datos basada en apuntadores pero con la diferencia de que esta guarda dos apuntadores, uno al siguiente nodo de la lista y uno al elemento previo. El poseer dos apuntadores nos permite recorrer la lista en ambos sentidos y permite un acceso bidireccional. (*“Doubly linked lists”*, 2023).

Debido a su característica de permitir acceso bidireccional a los datos dentro de la lista decidimos elegir la implementación de una lista doblemente ligada, ya que de esa forma el acceso a los datos de la bitácora será más rápido.

## **Algoritmos**

Para esta implementación se seleccionaron los siguientes algoritmos de ordenamiento y el siguiente algoritmo de búsqueda:

- Quick Sort : Complejidad  $O(n \log n)$
- Merge Sort: Complejidad  $O(n \log n)$
- Binary Search: Complejidad  $O(n \log n)$

### **Comparación entre algoritmos de ordenamiento**

Al momento de ejecutar el código con ambos algoritmos de ordenamiento no pudimos notar una diferencia en cuanto a los tiempos de ejecución ya que ambos algoritmos poseen la misma complejidad.

### **Referencias**

GeeksforGeeks. (2024, 23 enero). *Understanding the basics of linked list*.

<https://www.geeksforgeeks.org/what-is-linked-list/>

Codecademy. (s. f.). *Doubly linked lists*. Codecademy.

<https://www.codecademy.com/article/doubly-linked-list-conceptual>