

## Actividad 2.4

### Actividad Integral estructura de datos lineales

#### Hash table, stack & queue

#### Equipo:

Arturo Carballo Arias - A01662245

[a01662245@tec.mx](mailto:a01662245@tec.mx)

José Aram Méndez Gómez - A01657142

[a01657142@tec.mx](mailto:a01657142@tec.mx)

Juan Francisco García Rodríguez - A01660981

[a01660981@tec.mx](mailto:a01660981@tec.mx)

Programación de estructuras de datos y algoritmos fundamentales Gpo 631

David Alejandro Escarcega Centeno

23 de Octubre de 2022

**Realizar en forma individual una investigación y reflexión de la importancia y eficiencia del uso de las listas doblemente ligadas en una situación problema de esta naturaleza.**

Actualmente vivimos en una era con demasiada tecnología, y se maneja una enorme cantidad de datos en distintos ámbitos, por lo que siempre se tienen que encontrar formas muy eficientes para manipular y estudiar estos datos. Mediante la programación se han encontrado distintas formas posibles de llevar a cabo un orden ante tal cantidad de datos, como lo pueden ser los arreglos, las listas, pilas, o colas. Una de las estructuras de datos con la que hemos estado trabajando son las listas ligadas, las cuales han resultado tener algunas ventajas con respecto a otras estructuras como los arreglos; pero en este trabajo utilizamos listas doblemente ligadas, que básicamente es una variación de la lista enlazada individualmente.

Una lista enlazada es una estructura de elementos donde el orden de los mismos se establece mediante punteros, y la idea básica es que cada elemento de la lista incluya un puntero que indique donde puede encontrarse el siguiente elemento. Debido a estas características, el orden de los elementos puede ser fácilmente alterado modificando los punteros, lo que permite añadir o eliminar elementos; además de que no limita el número máximo de elementos. En las listas doblemente enlazadas, cada elemento de la lista está encadenado con el siguiente y el anterior; y al igual que en las listas sencillas, también tienen una cabeza y una cola, donde el puntero anterior de la cabeza se establece en NULL ya que este es el primer nodo, y el siguiente puntero del nodo de cola se establece en NULL ya que este es el último nodo. Si bien este tipo de listas ocupa más memoria, se justifica su uso en los casos donde es necesario recorrer la lista en los dos sentidos, y en listas donde se requiera borrar el último elemento, manteniendo complejidad  $O(1)$ .

Las listas doblemente ligadas son una de las herramientas más utilizadas en cuanto a la administración de datos se refiere. En el mundo real, estas son aplicadas en un diverso campo de situaciones, como por ejemplo:

1. *Visor de imágenes*: las imágenes anteriores y siguientes están vinculadas, por lo que se puede acceder a ellas mediante el botón Siguiente y Anterior.

2. *Página anterior y siguiente en el navegador web* : podemos acceder a la URL anterior y siguiente buscada en el navegador web presionando el botón Atrás y Siguiente, ya que están vinculados como una lista vinculada.
3. *Reproductor de música* : las canciones del reproductor de música están vinculadas a la canción anterior y siguiente. puede reproducir canciones desde el principio o al final de la lista.

Y también se utilizan en las ciencias de datos, para poder relacionar los datos con otros datos y que sean de fácil acceso al usuario o al software que deba procesarlos. Además que hacen más simple y eficaz la búsqueda de datos.

Para esta evidencia utilizamos muchas de las estructuras vistas hasta ahora durante este curso: como lo son las listas doblemente ligadas, los tablas hash, pilas y colas. Para ordenar se utilizaron dos pilas, las cuales dificultan un poco el ordenamiento de los datos, teniendo una complejidad de  $O(N^2)$ . Por último, para lograr la manipulación de las listas doblemente ligadas, se utilizaron funciones con complejidad de  $O(1)$  y  $O(N)$ .

**Aram:** Considero que lo más importante en programación es la eficiencia, esa es la premisa que engloba todo. Hacer cosas más rápido, más eficientes, con menos tiempo y memoria. Y el uso de estructuras de datos y memoria dinámica como la lista ligada nos lo permiten. Particularmente me encantan las listas ligadas por que son extremadamente eficientes al utilizar memoria, cuando en cambio estructuras como los arreglos utilizan memoria estática y que si fuéramos a hacer proyectos de este estilo con arreglos sería un desperdicio impresionante, además de que el código no es adaptable con otros datos. Me está gustando el curso por la implementación de los conceptos que hemos visto cómo templates, estructuras de datos, clases, sobrecarga de operadores, apuntadores, etc.

**Francisco:** En realidad pienso que estructuras como las listas doblemente enlazadas facilitan la representación óptima de los datos en la memoria de la computadora, sobre todo cuando la cantidad de datos que tenemos es realmente grande. Pienso que ordenar esa cantidad de datos utilizando pilas fue muy poco eficiente en tiempo, porque tarda demasiado tiempo el algoritmo en ejecutarse. Al final de cuentas pienso que usar un algoritmo de ordenamiento como quicksort es más útil para una cantidad de datos tan grande.

**Arturo:** Es vital el saber cómo emplear las listas ligadas, debido a lo eficiente que son, nos ayudan a administrar de mejor manera los datos, creando así arreglos con memoria dinámica, que facilitan el acceso a la información.