



# Tecnológico de Monterrey

## **Evidencia 2 - Actividad Integral de Estructuras de Datos Lineales (Evidencia Competencia)**

Mauricio Zavala Sánchez - A00837332

05 de octubre del 2023

### **Evidencia 2**

David Alonso Cantú Delgado

Programación de estructuras de datos y algoritmos fundamentales (Gpo 608)

El uso de estructuras de datos lineales, como Linked Lists, Doubly Linked Lists, Queue y Stack, es fundamental en informática y programación. Cada una de estas estructuras tiene su importancia y eficiencia en diferentes contextos, y a continuación, exploraré sus aplicaciones y ventajas individuales.

### 1. Linked Lists (Listas Enlazadas):

- Importancia: Las Linked Lists son una estructura de datos dinámica que permite la inserción y eliminación eficiente de elementos. Son fundamentales para la implementación de estructuras de datos más complejas, como listas, pilas y colas.

- Eficiencia: La inserción y eliminación en una Linked List son  $O(1)$  si se opera en un extremo (cabeza o cola), lo que las hace ideales para situaciones en las que es necesario un acceso rápido y constante a los elementos.

### 2. Doubly Linked Lists (Listas Doblemente Enlazadas):

- Importancia: Las Doubly Linked Lists son similares a las Linked Lists, pero con la ventaja adicional de poder navegar en ambas direcciones. Esto es útil en aplicaciones que requieren navegación inversa, como editores de texto.

- Eficiencia: Aunque ocupan un poco más de memoria que las Linked Lists estándar, proporcionan la misma eficiencia en términos de inserción y eliminación,  $O(1)$ , y permiten un acceso más rápido a los elementos vecinos.

### 3. Queue (Cola):

- Importancia: Una Queue es una estructura de datos que sigue el principio "primero en entrar, primero en salir" (FIFO). Es crucial para tareas de administración de recursos compartidos, como planificación de tareas en sistemas operativos y procesamiento de tareas en una cola de impresión.

- Eficiencia: Las operaciones de encolar (enqueue) y desencolar (dequeue) en una cola son  $O(1)$ , lo que la hace eficiente para su propósito.

#### 4. Stack (Pila):

- Importancia: Una Stack es una estructura de datos que sigue el principio "último en entrar, primero en salir" (LIFO). Es esencial para llevar a cabo operaciones como el seguimiento de llamadas de funciones en la pila de ejecución en la programación.

- Eficiencia: Las operaciones de apilar (push) y desapilar (pop) en una pila también son  $O(1)$ , lo que la hace eficiente y adecuada para su aplicación.

La elección de la estructura de datos correcta depende del problema específico que estés tratando de resolver. Cada una de estas estructuras de datos tiene sus ventajas y desventajas, y su elección afectará la eficiencia y el rendimiento de tu programa. Por lo tanto, es esencial comprender cuándo y cómo utilizar cada una de ellas.

En resumen, el conocimiento y la aplicación eficiente de estas estructuras de datos lineales son fundamentales en la programación y la informática en general, ya que permiten resolver una amplia gama de problemas de manera efectiva y optimizada.