



Tecnológico de Monterrey

Reflexión individual – Importancia y eficiencia de las listas doblemente ligadas

Franco De Escondrillas Vázquez | A01739410

Ingeniería en Tecnologías Computacionales

Dr. Daniel Pérez Rojas

TC1031.601

15 de octubre, 2025

Introducción

Para llevar el control y manipular un grupo de datos, es necesario almacenarlos en alguna estructura computacional. El modelo que se elija para desempeñar dicha función dependerá del tipo de datos, su cantidad y las operaciones que se desee realizar con ellos. En la actividad 2.3, se implementó una lista doblemente ligada para almacenar logs sobre fallos en el inicio de sesión y conexión al servidor desde diferentes computadoras. El volumen de datos abarca aproximadamente un cuarto de millón de elementos y las operaciones que se realizan con estos es la inserción, ordenamiento y búsqueda de elementos.

Para determinar si este tipo de estructuras lineales podrían ser las óptimas para el desarrollo de esta actividad específica, es necesario contemplar y analizar la complejidad algorítmica de cada una de las acciones.

Complejidad algorítmica de la lista doblemente ligada

Inserción al final | $O(1)$

Uno de los puntos fuertes de este tipo de estructuras de datos es que para agregar n elementos uno después del otro se necesita de una complejidad mínima $O(1)$ (tiempo constante). Sin embargo, para el resto de las operaciones que se realizan con los datos, es necesario tenerlos ordenados, por lo que en caso de utilizar este método sería indispensable utilizar además un algoritmo de ordenamiento.

Ordenamiento con mergesort | $O(n \log_2 n)$

Entonces, para obtener una lista de logs eficiente se puede implementar un mergesort, puesto que esta es la opción más eficiente en complejidad y más natural al trabajar con listas ligadas. Si se compara la complejidad temporal de estas dos opciones (inserción al final de n elementos y ordenamiento) con la de otras alternativas, como inserción ordenada, es evidente que esta elección es la mejor para una lista doblemente ligada.

En el primer caso, insertar n elementos requiere un tiempo de

$$n \cdot O(1) = O(n)$$

mientras que ordenarlos toma una complejidad mínima de $O(n \log_2 n)$, por lo que ambos combinados equivalen a

$$O(n) + O(n \log_2 n) \approx O(n \log_2 n).$$

Por otro lado, la inserción ordenada de n elementos tomaría un esfuerzo de

$$n \cdot O(n) = O(n^2),$$

que se vuelve peligroso para grandes volúmenes de datos.

Comparación gráfica entre ambos métodos

Inserción al final y mergesort $O(n \log_2 n)$

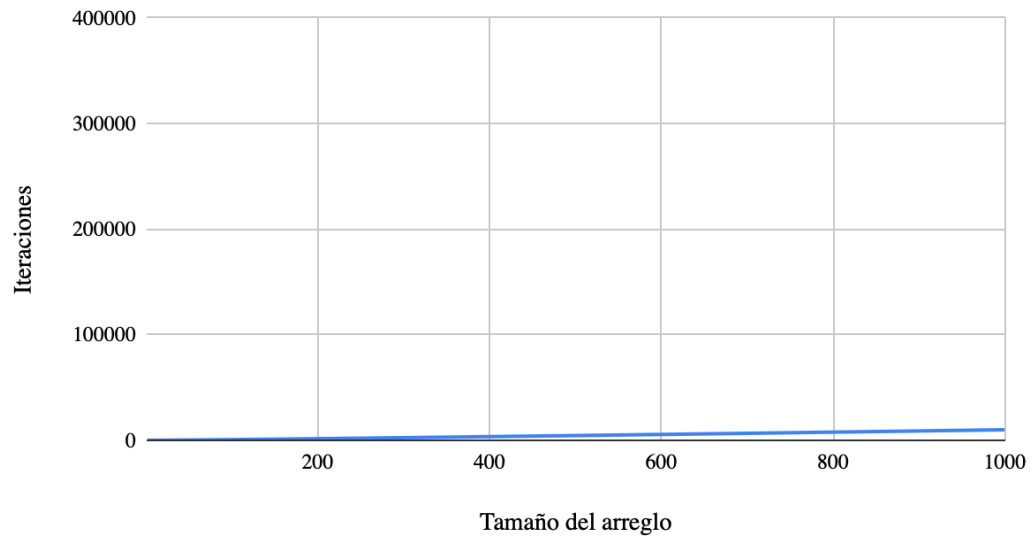


Figura 1. Complejidad inserción simple y mergesort

Inserción ordenada $O(n^2)$

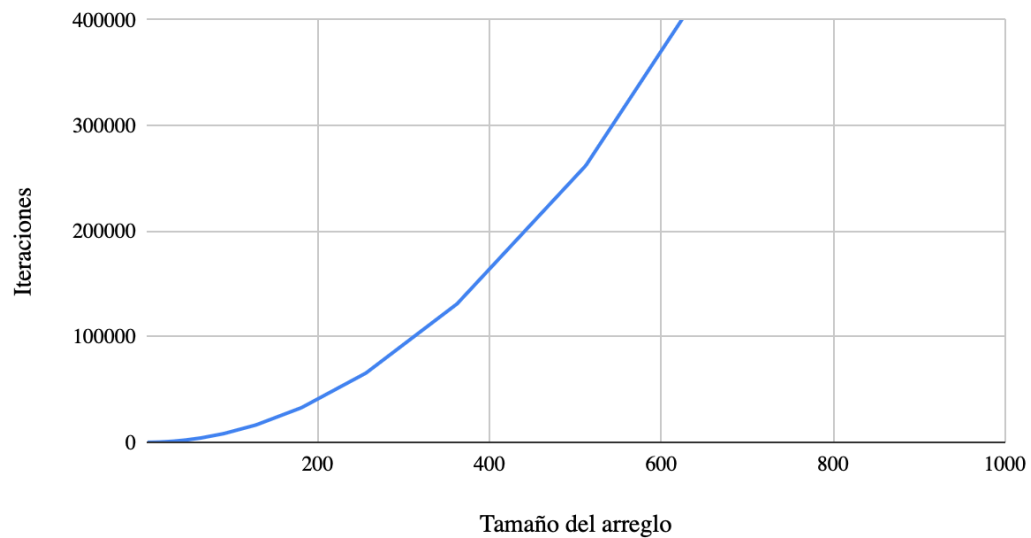


Figura 2. Complejidad inserción ordenada

Búsqueda secuencial | $O(n)$

Finalmente, la única forma con que se puede acceder a los elementos de una lista doblemente ligada es a través de los apuntadores al primer o último nodo (head y tail), según la versión implementada. Esto implica que, a pesar de que los elementos se encuentren ordenados, es necesario realizar una búsqueda secuencial.

En el mejor de los casos (si se calculara la mediana para determinar si comenzar a buscar por delante o detrás), la complejidad sería de $n \div 2$, que se redondea a lineal.

Búsqueda secuencial $O(n)$

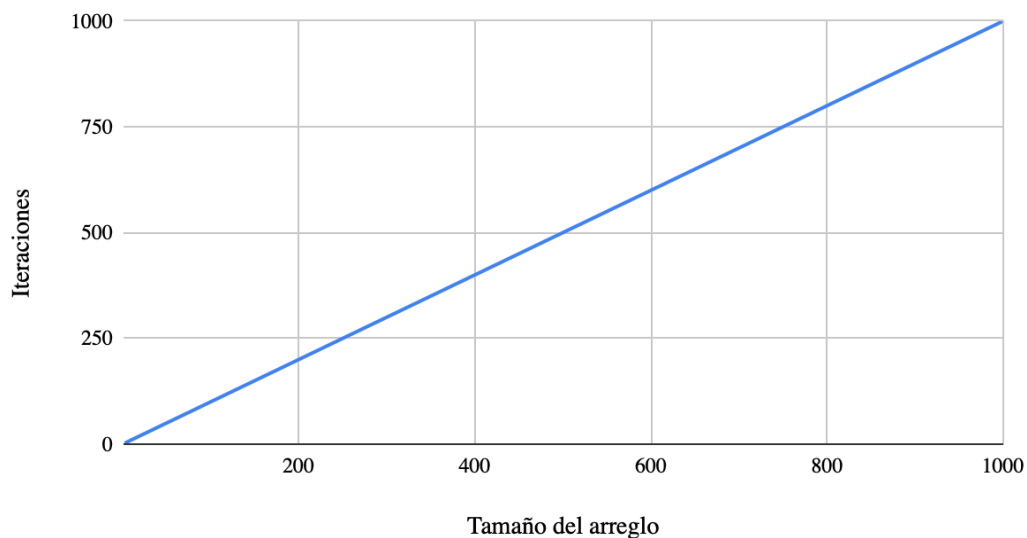


Figura 3. Complejidad búsqueda secuencial

Conclusión e impacto en la situación problema

Después de analizar los tiempos de complejidad, se puede decir que el uso de una lista doblemente ligada no causará problemas críticos si se trabajara con volúmenes de datos similares a los actuales. Sin embargo, esta tampoco es la óptima en este contexto. Si se compara con estructuras como árboles binarios, en donde la inserción ordenada de los elementos toma $O(\log_2 n)$ y la búsqueda de estos también se ejecuta en esta complejidad, entonces la velocidad de ejecución de la lista doblemente ligada queda totalmente opacada.

En el contexto de la situación problema (como se describió en una fase anterior del proyecto), la complejidad temporal es sumamente relevante, pues cuando se trabaja con big data y ciberseguridad el tiempo de respuesta ante posibles amenazas debe ser mínimo para garantizar la seguridad de una red, sus datos sensibles y los usuarios.