Act 2.3 - Actividad Integral estructura de datos lineales (Evidencia Competencia)

Marco Antonio García Mendoza
A01026487
Este código crea una especie de "lista de tareas" llamada DLinkedList, que permite realizar varias acciones, como agregar y quitar elementos, imprimir la lista, darle la vuelta y organizarla de diferentes maneras (usando Merge Sort y Quick Sort). La clase Registro representa una entrada en un diario y la clase Bitacora utiliza la lista de tareas para almacenar estas entradas y buscarlas en un período de tiempo específico.
El uso de DLinkedList se asemeja a una lista de tareas pendientes que puedes tener en tu vida diaria. Puedes agregar cosas, tacharlas cuando las completas, imprimir la lista, darle la vuelta para verlas en orden inverso, y ordenarlas de diferentes maneras.

La razón de la elección de una lista doblemente enlazada en lugar de una lista enlazada simple radica en su uso, es como si estuvieras organizando tu lista de tareas en una libreta, donde cada tarea es una hoja y necesitas hojas que se abran y se cierren, permitiéndote moverte hacia adelante y hacia atrás en tu lista de tareas. Esto facilita algunas operaciones, como darle la vuelta a la lista. Si solo pudieras hojear las páginas en una dirección, sería menos eficiente para ciertas acciones.

Las complejidades computacionales también se vuelven clave ya que como dice un artículo explicando la complejidad algorítmica “es una métrica teórica que nos ayuda a describir el comportamiento de un algoritmo en términos de tiempo de ejecución (tiempo que tarda un algoritmo en resolver un problema) y memoria requerida (cantidad de memoria necesaria para procesar las instrucciones que solucionan dicho problema).”
Las complejidades computacionales de algunas operaciones clave:
Inserción: O(1)
Borrado: O(n)
Búsqueda: O(n)
Actualización: O(n)
Inversión: O(n)
Ordenamiento: O(n log n)
En las ordenamiento tenemos 2: Merge Sort y Quick Sort para ordenar la lista. Aquí la más eficiente (rápida) fue “mergeSort” porque tardó 2,188,170 microsegundos mientras que “quickSort” tardó 109,189,050 microsegundos en ordenar la bitácora.

Referencias:
¿Qué es un algoritmo de ordenamiento? (2023). Platzi. https://platzi.com/tutoriales/1469-algoritmos-practico/6031-que-es-un-algoritmo-de-ordenamiento/
Medina, R. (2020, September 12). Estructura de Datos - Linked List (Lista Enlazada). DEV Community; DEV Community. https://dev.to/ronnymedina/estructura-de-datos-linked-list-lista-enlazada-2h9
Guillermo, J. (2018, June 23). ¿Qué es la complejidad algorítmica y con qué se come? Medium; Medium. https://medium.com/@joseguillermo_/qu%C3%A9-es-la-complejidad-algor%C3%ADtmica-y-con-qu%C3%A9-se-come-2638e7fd9e8c