

## ReflexAct2.3

Eugenio Peña García A01177348

Gracias a que realicé la evidencia Actividad 2.3 logré aprender mucho sobre el funcionamiento de las Linked Lists. Principal, y específicamente, aprendí como crear y utilizar las Doubly Linked Lists, las cuales son parecidas a las Linked Lists normales, excepto por el hecho de que los nodos poseen un previous y next en lugar de solo un previous, y almacenamos el valor de head y tail de la lista. Esto hace que crear las funciones de la lista sea más complicado, sin embargo, también nos da la opción de crear código mucho más eficiente, pues en lugar de tener que darle toda la vuelta a la lista cada vez que queramos acceder a algo, podemos simplemente comenzar desde el inicio o desde el final, dependiendo del caso.

Asimismo, aprendí a implementar distintos métodos dentro de esta Doubly Linked List. Por ejemplo, tuve que crear una forma de ordenar mis nodos dentro de una Linked List. Esto fue bastante retador, pero también me ayudó mucho a comprender ambos temas, pues necesitas entenderlos completamente antes de poder relacionarlos. Para hacer este ordenamiento (sort()), utilicé el método de MergeSort, con una complejidad de  $O(n * \log n)$ . Fue bastante complicado implementar este MergeSort a mi Doubly Linked List. Esto fue debido a que también teníamos que utilizar Queue o Stack dentro de dicho MergeSort. En este caso no tenía sentido utilizar Stack, por lo que me fui por el método Queue. Tuve que crear una clase completamente nueva para implementar este Queue, lo cual fue un reto porque Queue tiene sus propios métodos internos.

Esto me ayudó también a entender como funciona Queue y sus distintos métodos, principalmente Enqueue y Dequeue, los cuales estuve utilizando para el MergeSort constantemente. Al inicio tuve muchos errores debido a que no comprendía como funcionaba Queue. No sabía la diferencia entre rear y front de la lista, por lo que al hacer el Dequeue eliminaba el valor que agregué más reciente, y no el más viejo. Estuve horas buscando el error, pero una vez lo encontré, estoy seguro de que no me volveré a confundir en este tema.

Finalmente, pienso que el uso de las Linked Lists, principalmente la Doubly Linked List puede ser algo esencial en nuestra área. Tal vez hasta ahorita no había tenido necesidad de utilizarlas, sin embargo, para bases de datos, o cualquier programa que necesite almacenar listas de cualquier tipo, los Nodos y Linked Lists pueden ser las maneras más eficientes de organizar datos y crear listas. Esto lo comprobé con mi experiencia propia, aunque sea anecdotal. Mi primer programa (Evidencia 1.3) no me corría con todos los datos (16,807), solamente corría con hasta 1,000 datos. No obstante, ahora que utilicé Doubly Linked Lists para el programa, y lo único que cambió fue el uso de esto (y Queue), mi programa corre con los 16,807 datos y de manera rápida y eficiente, lo cual se me hace un cambio bastante significativo.