



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

Luz Patricia Hernández Ramírez A01637277

Programación de estructuras de datos y algoritmos fundamentales

Grupo 602

02 de octubre 2023

Act 2.3 - Actividad Integral estructura de datos lineales

(Evidencia Competencia)

Implementar una secuencia de nodos doblemente ligados dentro de una función buscador implica tener en consideración una variedad de factores sobre estructuras de datos. Especialmente en nuestro caso se tuvo que construir el nodo en base a lo que se estaba manejando como entrada de datos, es decir en base a archivos de texto. Nuestra estructura, por tanto, se definía con las características de la información de línea que fuera a leer el programa. Entre los atributos se establecieron las strings de mes, fecha, dirección ip y razón; Así como el entero día y los punteros del nodo: next y prev. La finalidad de esta estructura es que al momento de crear la clase de la lista, este orden preestablecido permite definir instancias que refieran al inicio y al final de la lista (head y tail) para facilitar el desarrollo de funciones que sean capaces de iterar sobre la lista y modificar su contenido sin necesidad de generarla desde cero, gracias a que este algoritmo trabaja accediendo a la memoria a través de punteros.

Para la búsqueda de IPs el método es práctico pues al permitir el ordenamiento específico de las direcciones no requiere hacer búsquedas exhaustivas; en lugar de buscar en toda la lista, simplemente se requiere buscar dentro de un rango específico, lo que mejora significativamente la eficiencia de la búsqueda.

Por otro lado, es importante tener en cuenta que la implementación de una estructura de lista doblemente ligada puede ser más compleja en comparación con otros algoritmos. Garantizar que el código esté correctamente diseñado puede requerir más atención, ya que es esencial gestionar adecuadamente los punteros y las conexiones entre nodos. Además, otro inconveniente notable es que consume más memoria en comparación con otras estructuras de datos, especialmente en listas largas.. Esto se debe a que cada nodo en la lista debe almacenar tanto los datos como los punteros a los nodos adyacentes, lo que puede aumentar el uso de memoria.

En lo que respecta a la complejidad computacional una vez ordenado y optimizado el archivo se puede reducir el tiempo sin embargo de primera mano todas las funciones implementadas en estos algoritmos tienden a la linealidad $O(n)$, pues a medida que la lista crece, el tiempo para realizar las operaciones también aumenta.

Referencias

Zaragoza, F. (2021). [pdf]. Algoritmos y estructuras de datos. *Universidad Metropolitana Unidad Azcapotzalco*. Recuperado de <https://academicos.azc.uam.mx/franz/aed/docs/dobles.pdf>