



## Reto. Reflexión Final

Carolina Ortega Barrios

A01025254

*Escuela de Ingeniería, Instituto Tecnológico de Estudios  
Superiores de Monterrey, México, México, Campus Santa Fe*

En este reto nos enfocamos a conocer y entender un poco de cómo funcionan los hackeos, cómo se pueden identificar y cuáles son las señales que podríamos considerar una “alerta roja” a la hora de revisar los contactos de ip que hemos tenido. Creo que es sumamente importante conocer los riesgos del hackeo y entender que todos estamos expuestos, ahora que estamos en la recta final puedo decir que aunque no llegue a comprender al 100% todo lo que implica el hackear a alguien más, durante el semestre empecé a comprender algoritmos que se pueden usar para identificar este fenómeno.

Al programar este reto tuvimos que tomar en cuenta que existen muchos tipos de estructuras de datos en las que cada una tiene su forma particular de organizar los datos y el objetivo es encontrar la manera más óptima, rápida y eficiente de encontrar datos en específico en archivos gigantes.

Al analizar el trabajo, llegue a la conclusión de que los grafos, los diccionarios y los vectores fueron de las estructuras que creo que más nos ayudaron. Los grafos nos ayudaron a conectar los datos mediante nodos que a su vez están conectados al dato específico que queríamos encontrar, nos ayudó muchísimo a poder leer estos datos, a dirigirlos por un camino y así visualizar las conexiones de una manera mucho más sencilla.

Por su parte los diccionarios nos ayudaron a asociar objetos mediante sus llaves y almacenar en ellos una cantidad grande de datos de la información que necesitáramos. También gracias al hashing nos fue posible identificar el momento en el que algo cambió en la página afectada (en nuestro caso “theguardian”) porque se ve claramente el comportamiento anormal en el día del hackeo.

Una de las cosas que sin duda se podrían mejorar es la complejidad del código, en el caso de mi equipo uno de los primeros pasos sería hacer uso de librerías como stl que tienen funciones que se podrían sustituir por funciones que implementamos nosotras. Al hacer esto, la reducción de la complejidad le permite al código funcionar más rápido y sin warnings (como nos pasó en nuestro código).

Aunque los grafos sirvieron de mucho, personalmente me costó un poco de trabajo comprender exactamente todos los datos que se imprimían porque a pesar de que depende de cómo se creó, como que no veía claramente entre tantos datos a la vez. Una de las cosas que más me ayudaron para poder identificar las conexiones hacia la página afectada fue una mejor organización del mismo grafo, en cuestión de ordenarlo con “: , endl , --->” para así tener una mejor visualización.