

# Reflexión Final

## Reto1:

Este reto me ayudó a refrescar sobre el sintaxis de C++, el uso de clases, los métodos de búsqueda y ordenamiento y el uso de vectores; además, aprendí a leer archivos y cargar los datos de este en un programa, para después manipularlos en la manera que más me convenga.

## Reto 2:

En este reto pude aplicar y comprender mejor el uso de stack y queue para poder organizar la información de la manera más rápida y eficiente posible, para manipular los datos y así poder encontrar las conexiones entrantes y salientes de una computadora específica.

## Reto 3:

Con este reto aprendí a idear mi propia técnica de hashing para poder insertar elementos en un arreglo; además, pude comprender las librerías de unordered\_map y list

## Reto 4:

Este reto me ayudó a comprender mejor los árboles binarios y reforcé mis conocimientos de unordered\_map del reto pasado.

## Reto 5:

Para el último reto con programación, juntamos principalmente el conocimiento del reto 2, 3 y 4 por el uso de unordered\_map, stack, queue y la manipulación de los datos al momento de ser cargados desde el archivo csv al código en C++, además de eso, aunque nos costó un poco el resolver el reto, pudimos comprender muy bien la manipulación de distintas estructuras de datos, que en si, estaban compuestas de otras estructuras.

## ¿Cuáles son las más eficientes?

Al finalizar este proyecto, creemos que las estructuras que nos resultaron más útiles para poder encontrar los problemas dentro de la red, fueron los grafos y los diccionarios. Ya que con los grafos, pudimos comprender muy bien la relación entre las computadoras dentro de la red; por otro lado, los diccionarios nos ayudaron mucho para poder observar el tráfico entre las computadoras de la red interna y las páginas web.

### ¿Cuáles podrías mejorar y argumenta cómo harías esta mejora?

Para mejorar nuestro programa modificaríamos nuestra función `top(int n, string date)` que regresa las top n sitios con más conexiones en date fecha para también poder utilizarla para sacar las n computadoras con más conexiones en date fech, así logrando encontrar a la computadora infectada por el botmaster. De igual manera, crearíamos una función que observará si un sitios tiene más de 50 conexiones entrantes más que el día anterior, puesto que en este caso significa que este sitio está obteniendo un tráfico anormal y lo más probable es que esté sufriendo un ataque de DDoS.