



Tecnológico de Monterrey

Reflexión actividad 5.2

Santiago Amir Rodríguez González

A01739943

Las tablas hash son estructuras de datos diseñadas para almacenar y recuperar información de manera extremadamente rápida. Funcionan aplicando una función hash a una llave para convertirla en un índice dentro de una tabla. Gracias a esto, las búsquedas, inserciones y eliminaciones pueden realizarse en tiempo constante, sin importar la cantidad de datos. Además, las tablas hash se utilizan ampliamente para crear bases de datos ligeras, gestionar accesos, clasificar información y construir sistemas donde la velocidad es fundamental.

También, las tablas hash son útiles porque permiten agrupar direcciones IP, contar accesos y detectar actividad rara sin tener que revisar toda la bitácora línea por línea y organizan la información de forma automática. Esto es súper importante cuando los datos llegan sin parar, como en servidores, plataformas digitales o redes de escuelas y empresas. En estos casos, las tablas hash no solo hacen el trabajo mucho más rápido, sino que también ayudan a encontrar patrones o problemas que, de otra forma, quedarían escondidos.

Cuando se analiza una bitácora de red, lo primero que pasa es que te encuentras con un montón de datos mezclados. Si no los organizas, todo se vuelve un caos. Ahí es donde las tablas hash nos ayudan a resolver problemas ya que: permiten agrupar automáticamente la información por redes, que es la forma más lógica y útil de ordenar las IPs. Básicamente, te dan acceso rápido a cada grupo sin tener que peinar toda la bitácora cada vez.

Lo mejor es que el programa no necesita revisar línea por línea para encontrar la información de una red. Con la función hash, llegas directo al lugar donde está el resumen. Esta diferencia se nota muchísimo cuando la bitácora empieza a crecer con los días y llega a tamaños que serían imposibles de manejar con estructuras más básicas.

Además de la velocidad, las tablas hash hacen que todo se vea más claro. Agrupar IPs, contar accesos o identificar hosts únicos deja de ser un lío porque cada red tiene su “espacio” dentro de la tabla. En situaciones reales esta capacidad de ordenar y consultar rápido la información puede marcar la diferencia al momento de detectar ataques, fallas o simplemente entender cómo se usa la red.

El hecho de que puedas convertir una dirección IP en una llave que te lleva directo al resumen de su red demuestra lo potente que puede ser un buen diseño de datos. En vez de perder tiempo buscando entre líneas, una tabla hash funciona como una guía telefónica automática que te lleva justo al lugar que necesitas. Esto no solo acelera el análisis, también cambia la forma en la que ves la información: ya no es un montón de direcciones sueltas, sino un conjunto de patrones que tienen sentido.

En cuanto a SHA-256 dentro del contexto del proyecto, es importante reconocer que esta función hash es muy robusta, pero también pesada para tareas que se ejecutan miles de veces por segundo, como procesar direcciones IP en una tabla hash. Su complejidad computacional es mayor y eso puede afectar el rendimiento. Tiene ventajas claras, como su resistencia a colisiones y su uso común en seguridad digital, pero no es la alternativa ideal cuando se busca velocidad pura. Aunque puede aplicarse en áreas como firmas digitales, autenticación o verificación de integridad, para organizar accesos de red es mejor usar funciones hash más ligeras y específicas para estructuras de datos.

Y al final, usar tablas hash te obliga a pensar en rendimiento desde el principio. Elegir el tamaño correcto, manejar colisiones y preparar la tabla para que pueda crecer son decisiones que influyen directamente en qué tan buena será la aplicación. Este tipo de proyectos no solo te enseñan a programar, sino a crear soluciones que realmente pueden escalar y enfrentarse al tipo de datos que hoy mueve al mundo.

Referencias:

<https://learn.microsoft.com/es-es/powershell/scripting/learn/deep-dives/everything-about-hashtable?view=powershell-7.5>

<https://learn.microsoft.com/es-es/dotnet/api/system.security.cryptography.sha256?view=net-8.0>