

### **Reflexión Actividad 5.2: Actividad Integral sobre el uso de códigos hash**

En esta actividad se implementó la estructura de datos *unordered\_map* de la STL de C++ que funciona como una hash table. Una hash table es una estructura que asocia llaves únicas o “keys” de cierto tipo de dato, que asocian a otro tipo de dato o a un objeto, u otra estructura como fue el caso en esta actividad 5.2. Las llaves de un *unordered\_map* no pueden repetirse, es decir, deben ser únicas en la instancia de la estructura. Esta es una de las características que hacen que este objeto sea especial.

Algo particular que hace que esta estructura destaque, es que no están ordenados los valores que se insertan a la hash table, y que los métodos básicos como lo son lectura, búsqueda y eliminación son  $O(1)$ . Las hash tables reciben su nombre por la función hash con la que opera la tabla. Una función hash es un algoritmo que, dado un input, con simples operaciones aritméticas determinará una posición en la tabla que apunta a un espacio desocupado y ahí insertará, buscará, leerá o eliminará los datos. Dado un input, se espera siempre un mismo output (esto puede cambiar dependiendo del nivel de complejidad de la función hash), por lo que el uso de la llave correcta para acceder al valor esperado es de suma importancia.

Para esta actividad, se creó un *unordered\_map* que reciba de key un string, que en este caso es una IP recuperada de los logs de la bitácora. El tipo de dato al que permite el acceso una llave es uno de tipo *IpAddress*, donde se guardan los datos de cada log de la bitácora como atributos. Los atributos son: fecha, hora, IP, mensaje del log y una variable que cuenta la cantidad de veces que una IP se repitió en todo el archivo de la bitácora. Recordemos que no existen llaves repetidas en un hash table como la que se creó aquí, por lo que, si un valor de llave vuelve a aparecer, se incrementa la variable contador en 1 del objeto *IpAddress*.