



**Programación de estructura de datos y algoritmos fundamentales**

Actividad 5.2 Actividad Integral

Sebastian Antonio Almanza A01749694

Tecnológico de Monterrey

6 de Febrero del 2024

Grupo 570

Profesor: Eduardo Arturo Rodriguez Tello

Una tabla Hash es una estructura de datos en la cual se utiliza un método de Hashing para asignar a cada elemento una posición en la tabla. La tabla Hash tiene como ventajas el poder permitir que operaciones como encontrar y agregar datos no sean operaciones de mucha complejidad. Una complicación de las tablas Hash es cuando la llave de inserción se repite en más de un elemento provocando una colisión, es ahí donde se debe de tener un método para manejar las colisiones de elementos, en las tablas Hash existen dos métodos principales de manejo de colisiones las cuales son:

- **Direccionamiento Abierto:** Este método consiste en “reacomodar” de manera interna el elemento en una celda no ocupada. Un ejemplo es el método de **Prueba Lineal** en el cual se recorre la tabla hasta que se encuentra una celda no ocupada donde alojar el elemento colisionado.
- **Encadenamiento:** Este método se basa en utilizar una parte externa de la tabla para acomodar el elemento colisionado. Un ejemplo puede ser el método de **Encadenamiento Externo** en el cual se utilizan listas ligadas para almacenar elementos colisionados de la misma posición de la tabla Hash.

Para nuestra implementación decidimos utilizar una tabla Hash ya que realizar operaciones de búsqueda en ella es bastante eficiente dada una IP ya que en nuestra tabla Hash estaría contenido un resumen de las Ips con su contenido (grado de entrada/salida, conexiones a otras Ip). Los principales métodos de nuestra implementación son:

- **construyeHash():** Este método nos ayuda a construir una tabla Hash que contenga como llave la Ip en decimal y como dato la información total de una Ip, ocupa la prueba lineal para el manejo de colisiones. Tiene complejidad de  $O(1)$ .
- **getIpSummary():** Este método pide al usuario una Ip con la cual realizar una búsqueda en la tabla Hash para obtener los grados de entrada y salida, y con ayuda de la lista de

adyacencias obtener la lista completa en orden descendente de las ips con la cual está conectada la ip dada por el usuario. Tiene complejidad de  $O(n \log n)$ .

En nuestra implementación también fue importante conocer las colisiones y el factor alfa ya que si estas son demasiadas comienza a haber problemas en la asignación de espacio en la tabla Hash, lo que puede incrementar la complejidad de operaciones como búsqueda o la inserción de nuevos elementos. En nuestro caso obtuvimos que hubo 9637 colisiones y un factor alfa de 0.666135, lo cual nos indica que logramos manejar de forma efectiva las colisiones mediante el uso de un número primo para el tamaño de la tabla Hash.

### **Referencias**

*Metodo de dispersion (Hashing) en C++*. (2013, 4 marzo).

<https://blog.martincruz.me/2013/03/metodo-de-dispersion-hashing-en-c.html>