

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
PROGRAMACIÓN DE ESTRUCTURAS DE DATOS Y ALGORITMOS
REFLEXIÓN

TC1031.501: PROGRAMACIÓN DE ESTRUCTURAS DE DATOS Y ALGORITMOS (GPO 501)

Michelle Andrea Arceo Solano

Tecnológico de Monterrey, Campus AGS

a01625268@itesm.mx

Liga del repositorio: <https://github.com/MichelleArceo/TC1031.501/tree/main/Actividad5.2>

Las estructuras de datos sin relación son los que conocemos como conjuntos, por lo que comúnmente se utilizan arreglos para representar la distribución y estas estructuras almacenan elementos simples como compuestos (objetos) por lo que la forma en la que se almacenan los elementos está definida por la técnica de Hashing que se utilice. [1]

Se les llaman técnicas de hashing en las formas en las que se dispersan los elementos que se ingresan a las funciones de hashing, estas técnicas se preocupan por dos cosas [1]:

- Por el manejo de colisiones ya que no pueden asegurar que se garantice un valor distinto por cada dato.
- Poder distribuir los datos lo más aleatorio posible en el arreglo para que se pueda ejecutar el programa de forma eficaz.

En la siguiente tabla podemos observar la complejidad de métodos más comunes en el diseño de las tablas hash:

Operaciones	Eficiencia (Peor de los casos)
Sondeo cuadrático [2]	La complejidad es de $O(n)$ en el peor de los casos ya que si se insertan muchos elementos en la misma clave puede llevar tiempo o en el caso de que supere de equilibrio de carga, tienen que hacer una tabla más grande y tener que volver a hacer todo el proceso de insertar datos. En casos promedio llega a ser $O(1)$ ya que no es común que elementos tengan la misma clave.
Encadenamiento [2]	

Estos tipos de estructuras de datos y búsqueda tiene varias aplicaciones como en los software de computadora, indexación de bases de datos, enfocados en las matrices asociativas, cachés y en su especialidad, los conjuntos. Aparte ofrece recuperación de datos más flexible y confiable que otras estructuras, lo único malo es que no es muy bueno ordenando datos ya que como se explicó anteriormente funciona de forma aleatoria para ejecutar el programa rápidamente. [1]

Las tablas hash son excelentes por la velocidad que tienen para distribuir los datos en el arreglo, esto tiene bastante ventaja por encima de las demás estructuras cuando el número de entradas son miles como es el caso del archivo de entrada de esta actividad (bitácora3.txt), por lo que se ajusta perfectamente en la implementación de este programa y si se tiene con anticipación la cantidad de datos es mucho mejor ya que la tabla nunca cambia de tamaño y distribuye todos los datos.

REFERENCIAS

- [1] Drozdek, A. (2012). *Data Structures and algorithms in C++*. Boston, United States of America. Cengage Learning.
- [2] Allen, M. (2013). *Data Structures & Algorithm Analysis in C++*. Florida, United States of America. Prentice Hall.