Tecnológico de Monterrey

Programacion de estructuras de datos y algoritmos fundamentales

TC1031

Reflexión Actividad 5.2.

Profesor: Daniel Pérez Rojas

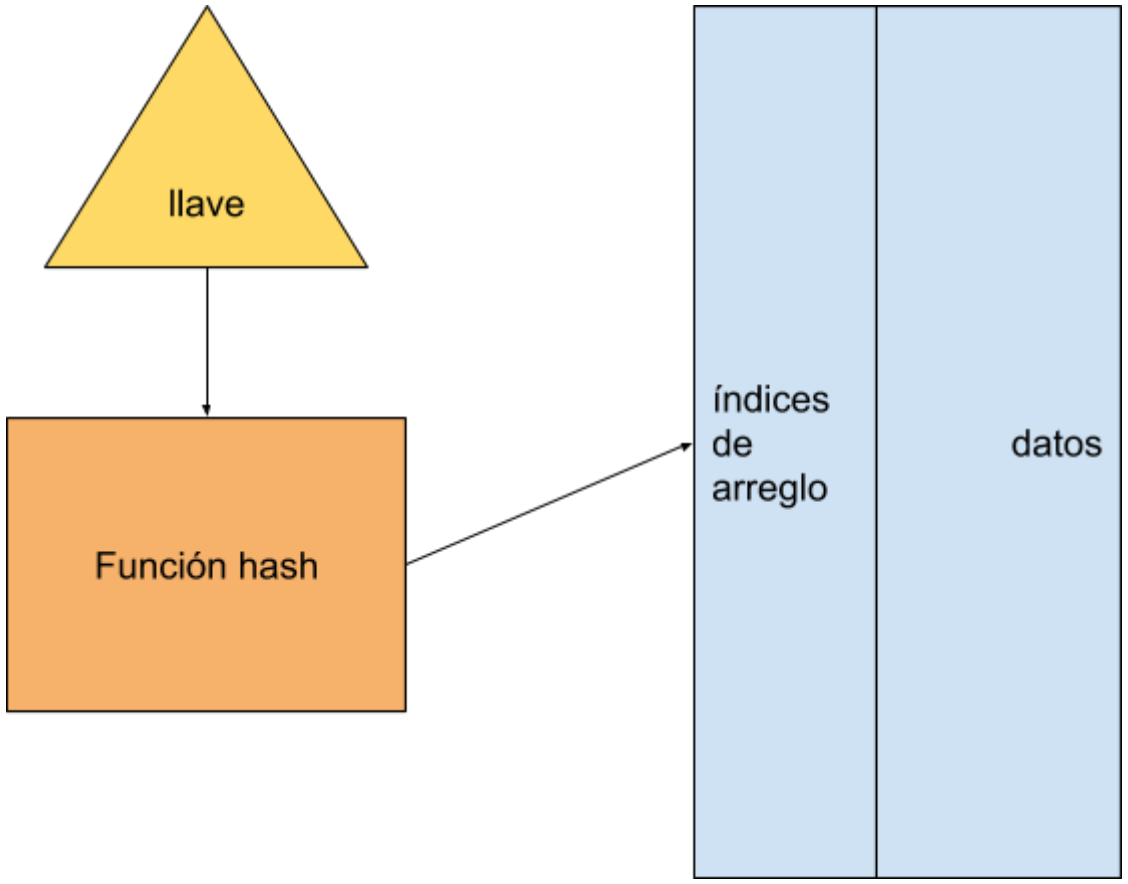
Alumna: Alexa Basurto Flores

Matricula: A01422793

Campus: Cuernavaca

Fecha de entrega: 29/11/2020

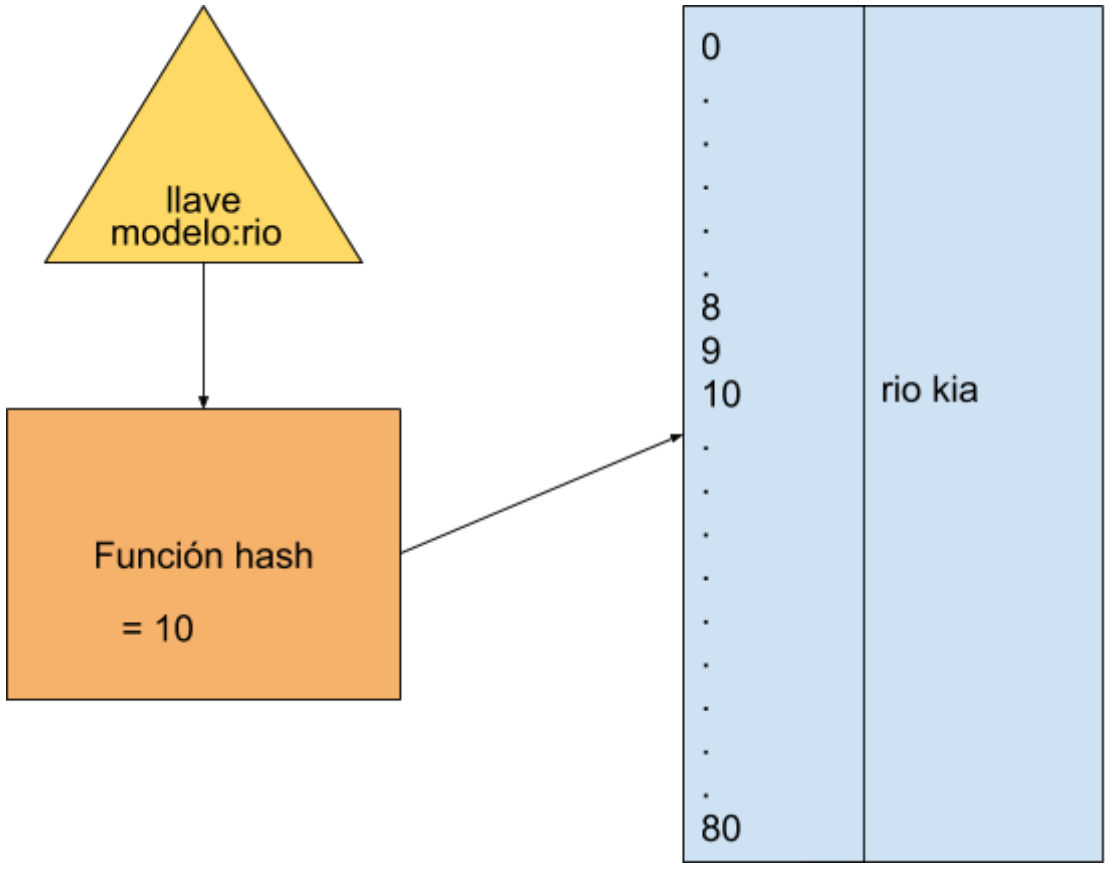
Una Tablas Hash es una estructura de datos que puede guardar datos de una forma lineal en un arreglo, utilizando una llave para insertar datos a dicha tabla.
Cuando se quiere insertar un dato al arreglo se debe tener un arreglo de cierto tamaño,declarar una llave y hacer la función hash para obtener el índice en donde se podría introducir el dato en el arreglo.



Un ejemplo de lo anterior podría ser que tenemos un arreglo de tamaño 80, queremos insertar datos de modelos de carros como {rio kia,niro kia,sorento kia} como podemos ver tenemos el modelo y marca de carro pero declaramos como llave el modelo, una vez haciendo esto tenemos que sumar los valores ASCII de cada letra del modelo de esta manera para modelo rio la suma de sus caracteres ASCII seria 114 + 105 +111 = 330 y a este resultado le aplicamos un % por el tamaño del arreglo que es 80 lo cual nos da el residuo de lo que sería la división de 330/80 el residuo de lo anterior es 10

```
>>> ord('r')
114
>>> ord('i')
105
>>> ord('o')
111
>>> 114 + 105 +111
330
>>> 330%80
10
```

este método de hash se tendrá que hacer para cada uno de los modelos ya que son nuestras llaves, teniendo el valor hash que en el caso del modelo río es 10 nos indica en qué índice del arreglo se van a insertar los datos que en este caso son el modelo y marca de carro.



Ahora no siempre se tiene que utilizar esta función de hash de sumar cada caracter, existen otras formas de hacer esto ya sea por medio de selección de dígitos en donde se seleccionan unos dígitos de la llave para insertar sus respectivos datos,por residuales que es lo que se vio en el ejemplo anterior donde se suman los caracteres y se obtiene el residuo de esto por el tamaño de dicho arreglo, cuadrática donde elevamos los datos de la llave al cuadrado y tomas los datos centrales para ver en donde insertar en la tabla y folding donde sumas caracteres y aplicas alguna función matemática para obtener el índice.
Debido a que hay números de caracteres que se repiten se llegan a tener colisiones a la hora de insertar una dato, esto significa que dos llaves generan el mismo valor y el segundo dato en querer entrar al arreglo se topa con el dato previamente insertado por lo que se dice que colisionan, cuando esto sucede se puede resolver de dos métodos :

- método de encadenamiento o de hashing abierto
  donde se trabajan con listas encadenadas para realizar la reacomodación de los datos que colisionan.
- método de dirección abierta o de hashing cerrado
  donde de forma iterativa se aumenta el valor que dio la llave para que se inserte un o varios índices después, en este método cuando se llega al final del arreglo se va hasta el principio para ver si hay un lugar disponible.

Teniendo todo esto en mente se sabe que la complejidad de tiempo de una tabla de hash bien elaborada toma una complejidad en O(n) lo cual es lineal que significa que es rápido en cuanto al manejo de datos grandes y la búsqueda de datos en un arreglo grande, pero uno de los problemas principales es que surgen muchas colisiones si no se usa una función de hash eficiente.

Las tablas de hash se usan en varios proyectos que se necesita buscar datos cuando son muchos de estos, un ejemplo de esto es la función de hash de SHA-256 que se utiliza para encriptar datos como bitcoin, donde se tiene una llave y se obtiene un hash con mayores caracteres que la llave pero si se hace un ligero cambio en la llave se obtiene un hash completamente diferente por lo que lo hace difícil de descifrar ya que son demasiados caracteres para poder decodificarlo.

La ventaja de este método de hash es muy beneficiario en cuanto encriptación y velocidad porque SHA-256 es rápido cuando son datos grandes y cuando son datos pequeños son mucho más rápidos, y la desventaja de esto es que se predice que en un futuro se pueda descifrar esta encriptación de manera más sencilla porque actualmente existen conceptos de decodificación pero no se necesitan varias máquinas para decodificar.

Fuentes:

- *Tablas hash: acceso rápido a valores hash desde la base de datos*. (2020, October 12). IONOS Digitalguide. https://www.ionos.mx/digitalguide/servidores/seguridad/tablas-hash/
- EcuRed. (n.d.). *Tabla hash - EcuRed*. Ecu Red. Retrieved November 29, 2020, from https://www.ecured.cu/Tabla_hash
- Maldonado, J. (2020, April 28). *¿Qué es SHA-256? El algoritmo criptográfico usado por Bitcoin*. Cointelegraph. https://es.cointelegraph.com/explained/what-is-sha-256-the-cryptographic-algorithm-used-by-bitcoin