**Hashing**

1. El hashing es una técnica que nos permite acceder rápido a registros de un archivo por medio de claves, con un único acceso a disco. Permite generar una dirección única en base a la clave que se recibe, para su almacenamiento.
2. La **función de dispersión** se puede entender como una “caja negra”, que recibe por entrada una clave y la transforma para poder obtener una dirección base donde se va a encontrar el registro con dicha clave. Esta función permite la dispersión de los registros de la forma más uniforme posible, dentro de un rango posible.
   1. ***1er Funcion:***
   2. ***2da Funcion:***
   3. ***3ra Funcion:***
3. Conceptos:
   1. **Sinónimo:** cuando claves diferentes resultan en la misma dirección.
   2. **Colisión:** cuando las claves sinónimo resultan en la misma dirección, ambas la quieren ocupar pero no hay lugar para ambas.
   3. **Overflow:** luego de aplicar hash, una clave se desea almacenar en una dirección que ya se encuentra ocupada.

Para que en un archivo ocurra una colisión y no un desborde, este debe contar con algún método para resolver esto:

* Poseer un algoritmo de dispersión **perfecto** (muy dificil), capaz de contemplar las colisiones y brindar ubicaciones diferentes.
* Reducir el número de colisiones al mínimo para poder tratarlas:
  + Distribución aleatoria de direcciones.
  + Mayor cantidad de espacio (espacio obsoleto, pero menos colisiones).
  + Almacenar más de un registro en una misma dirección. Esto limita la cantidad de registros que puede almacenar cada “nodo”

1. Respondida en el punto anterior.
2. La **densidad de empaquetamiento** es la relación que existe entre el número de registros que componen un archivo y el espacio disponible que hay para almacenar dichos registros. A menor DE, se beneficia la posibilidad de tener una mayor dispersión (menor probabilidad de colisiones), pero al haber más espacio, hay más espacio inutilizado.
3. Técnicas:
   1. **Saturación Progresiva:** permite almacenar un registro en la dirección próxima disponible de la colisión. Es decir, la más cercana, pero esto produce una saturación sectorizada, por lo que todos los registros, en caso de saturación, se esparcen a nodos cercanos.
   2. **Saturación progresiva encadenada:** similar al caso anterior, pero dada una saturación, la dirección anterior mantiene un “enlace” con la nueva dirección, permitiendo una secuencia al momento de la búsqueda.
   3. **Saturación progresiva encadenada con área de desborde separada:** dado que ante un caso de colisión, se re-direcciona a una nueva dirección, esto provoca que los registros no estén en la dirección original. Por lo tanto, se propone un sector reservado para almacenar estos registros en colisión, por lo que la función de dispersión no puede alcanzarse. La dirección original apuntará a la dirección correspondiente en este sector.
   4. **Dispersión doble:** por medio de 2 funciones, obtiene en la primera la dirección original, y en caso de desborde, obtiene un desplazamiento por medio de la 2da. Así provoca que se esparzan los registros a lo largo del archivo.