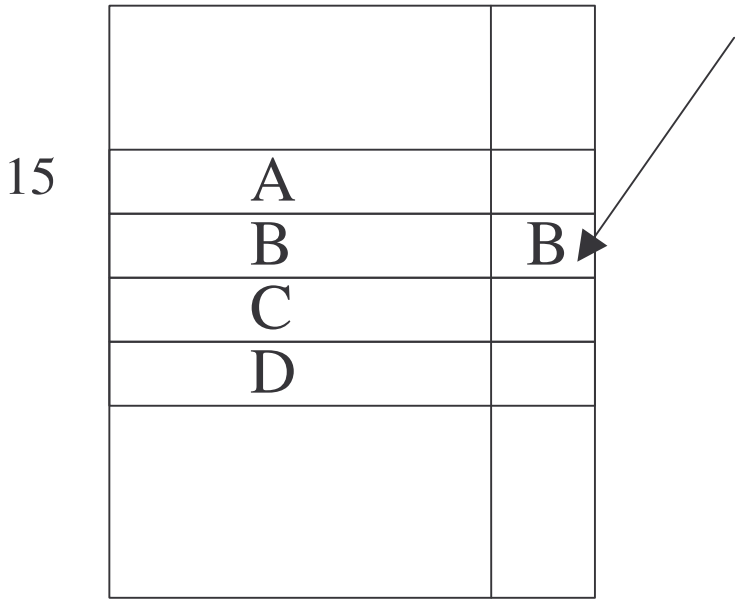
Archivos Directos

Resolución de Colisiones

Abierto Lineal Cíclico

- El registro es ubicado en el primer lugar desocupado a partir de la dirección obtenida por la función de hashing. Cuando se llega al fin del archivo se continua a partir del primer lugar del mismo.



Estados posibles de los registros:

Ocupado, Borrado, Libre

F(A)=15

f(B)=15

f(C)=15

f(D)=16

Escribir A, B, C, D

Borrar B

Leer D

Abierto Cuadrático Cíclico

- Similar al Abierto lineal cíclico, solo que los registros sinónimos son ubicados en el archivo cada una cierta cantidad de registros y no linealmente.
- Evita generar zonas del archivo superpobladas.

Abierto Random Cíclico

- Similar a los métodos lineales y cuadráticos.
- Las colisiones se ubican en una posición determinada por una función pseudo-aleatoria, cuya semilla es la posición donde se produjo la colisión.
- Evita el problema de las zonas superpobladas y desiertas, lo cual reduce la probabilidad de que se produzca una colisión.

Doble Hashing

- La posición a utilizar para las colisiones esta dada por una segunda función de hashing.
- Si esta segunda posición también estuviera ocupada, podría continuarse en forma lineal, cuadrática, random, o usando otra función de hashing.

Abierto con Area de Overflow

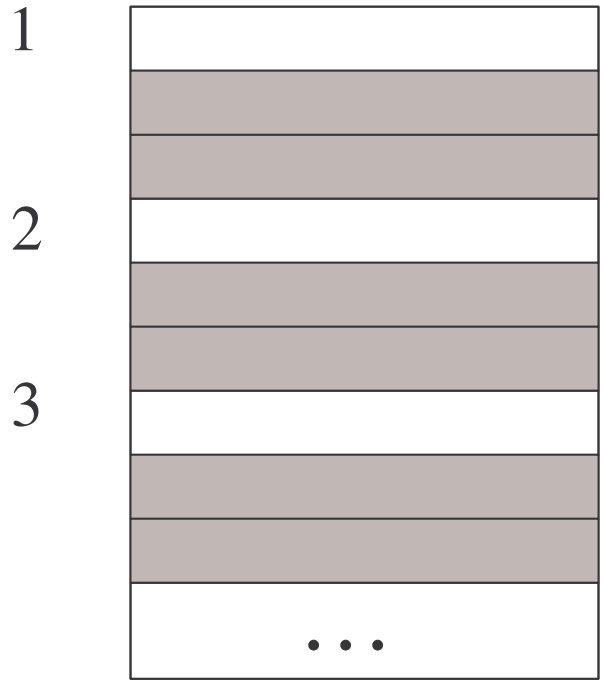
- Los sinónimos no se ubican en un lugar “común” del archivo, sino que se colocan en un area especial denominada zona de overflow.
- El área de overflow puede estar ubicado en el mismo archivo o en un archivo aparte.

Abierto lineal/ cuadrático/ cíclico con área de overflow

- Estos métodos son iguales a los anteriores, pero la ubicación de los sinónimos se hace dentro del área de overflow.
- El manejo del área de overflow dependerá del método elegido.

Area de Overflow distribuida

- El área de overflow se ubica distribuida entre todo el archivo de datos.
- Optimiza los accesos.



Ejemplo:
2 registros de overflow por cada uno de datos.

Siendo:

a= Registros de datos

b =Registros de overflow

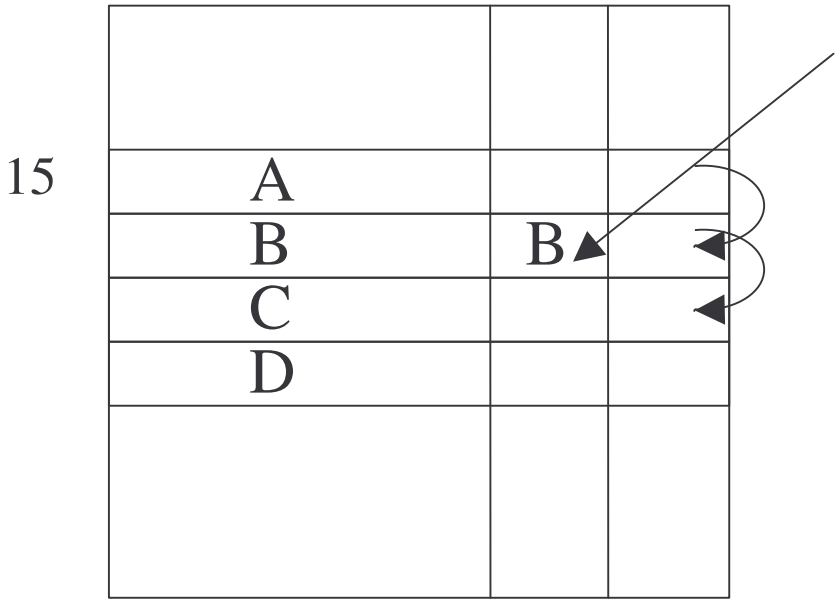
Posdato(x) = x + b (x div a)

Posovf(x) = b (x div a) + a [(x div a) + 1)]

Direccionamiento Cerrado sin área de Overflow

- Los sinonimos son encadenados en el archivo, formando una lista de sinonimos.
- Los mismos se pueden ubicar en el archivo segun cualquiera de los metodos antes vistos para direccionamiento abierto.
- En el caso de manejar área de overflow, la misma podrá implementarse con cualquiera de estos métodos, ya sea en el mismo archivo como en archivos separados.

Ejemplo: Lineal cerrado sin área de overflow



Estados posibles de los registros:

Ocupado, Borrado, Libre

F(A)=15

f(B)=15

f(C)=15

f(D)=16

Escribir A, B, C, D

Borrar B

Leer D

Buckets

- Arreglo de registros en el archivo, en donde se pueden almacenar los sinónimos.
- La entrada/salida se realiza a nivel de bucket.
- Simplifica el manejo de sinónimos en el archivo
- Obliga a reservar espacio para los sinónimos, que probablemente no será utilizado.
- Cuando los buckets se llenan, podrá utilizarse alguna de las técnicas antes descriptas.

Hashing Extensible

- Permite un crecimiento del archivo sin cambiar la función de hashing.
- Utiliza una función de hashing dinámica, una tabla de alocación de buckets y un archivo de datos compuesto por una cierta cantidad de buckets siendo posible agregar más de ser necesario.
- Función de hashing dinámica: devuelve un número entero de b bits. En principio, solo los I primeros bits serán utilizados para direccionar los datos.

Ejemplo de hashing extensible

Vamos a comenzar considerando la inserción de 5 registros, y tomando los dos primeros bits de la clave.

F(r1) = 00...
F(r2) = 01...
F(r3) = 10...
F(r4) = 11...
F(r5) = 11...

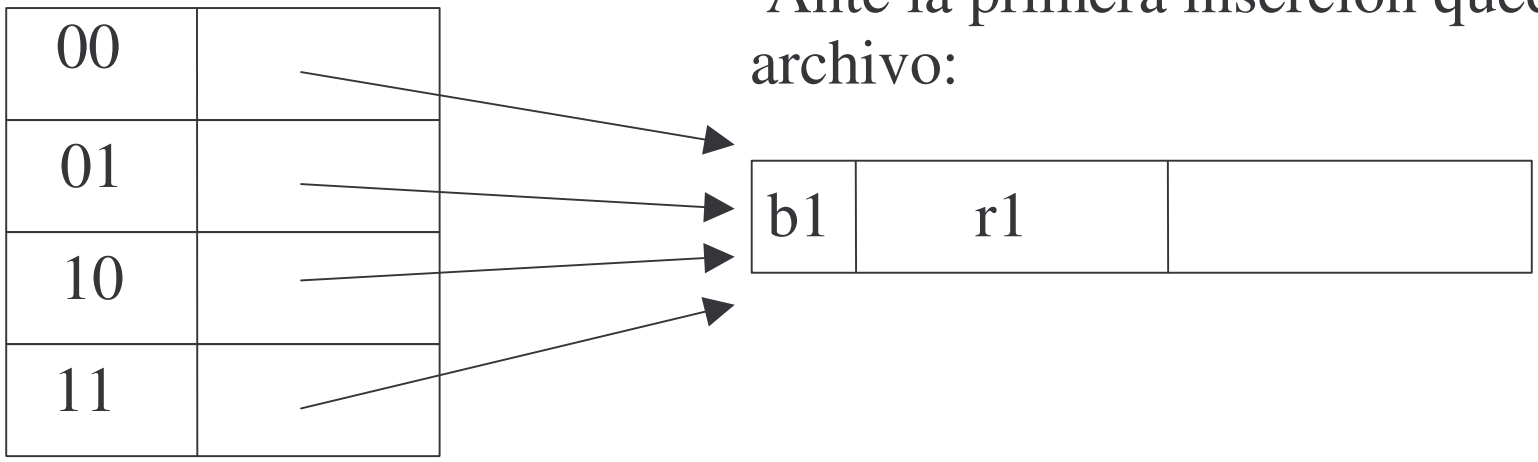
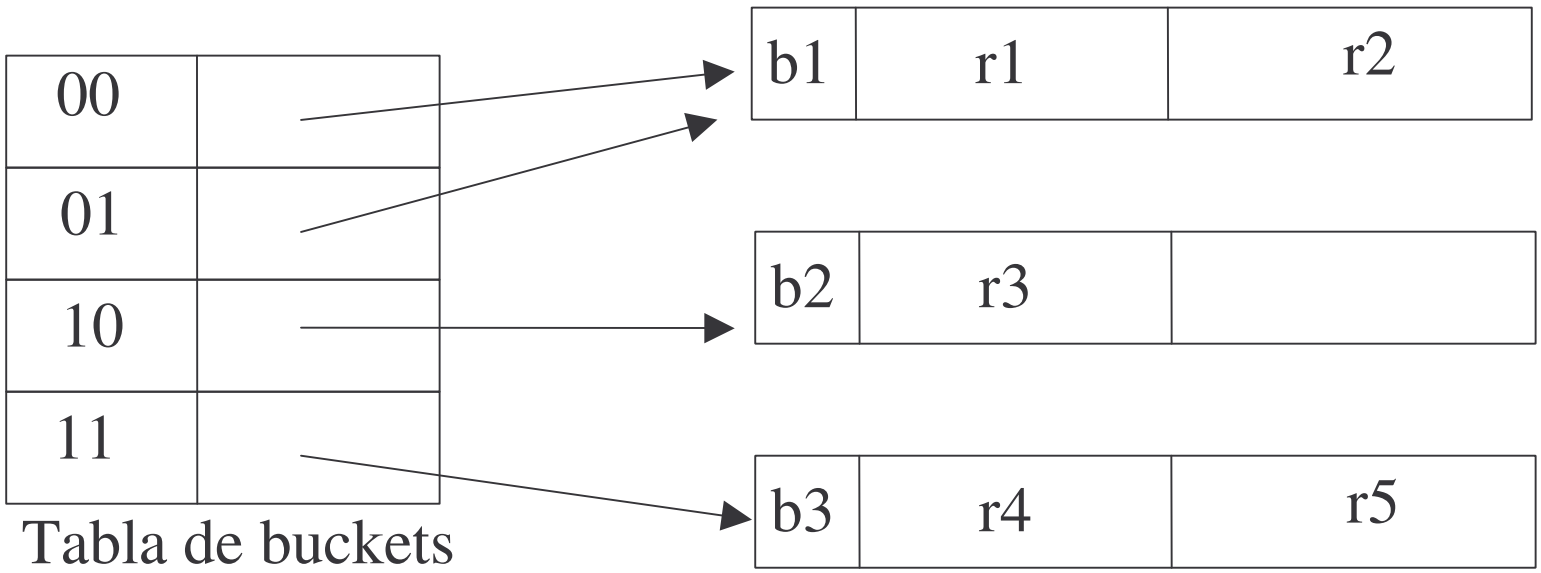


Tabla de buckets

Ante la primera inserción quedaría el archivo:

Ejemplo de hashing extensible

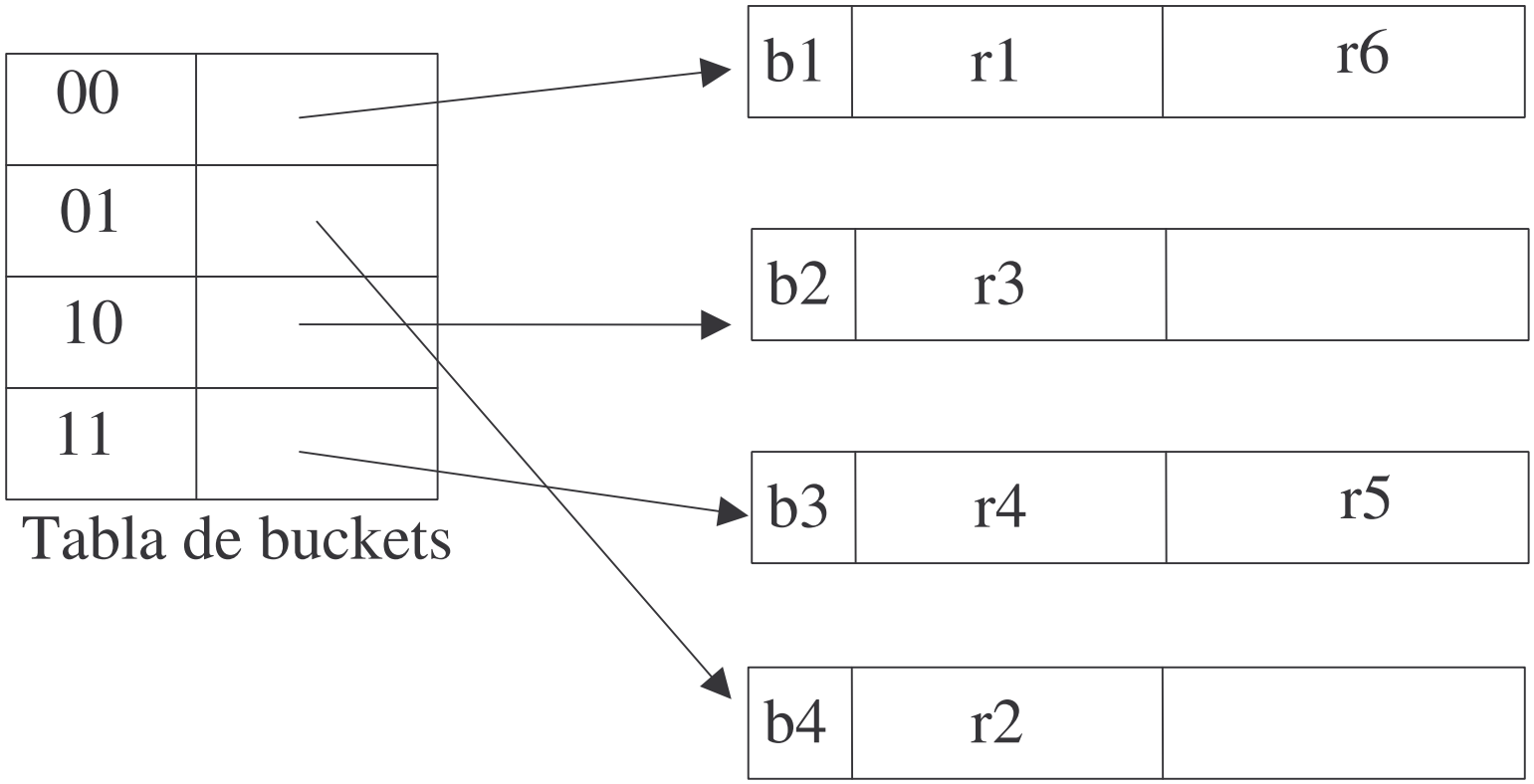
Una vez realizadas las 5 primeras inserciones el archivo queda:



Suponemos entonces que necesitamos ahora agregar un registro en el bucket b1 o en el b3.

Ejemplo de hashing extensible

Agregamos ahora r6 en b1 (f(r6) = 00...):



Ejemplo de hashing extensible

Agregamos ahora r7 en b3. En este caso también incrementamos I: