

## Documentación Hash

Se implemento dispersión estática. Los desbordes se resuelven usando direccionamiento abierto y doble dispersión. Se usa una funcion para obtener la direccion base y otra para dar los saltos en caso de desborde.

### Funciones:

**Función principal:** Resto de División o Módulo . La clave se divide entre el número de direcciones . Para asegurarnos una dispersión eficiente nos aseguramos que el tamaño de la tabla sea un numero primo.

**Función secundaria:** Tambien es Resto de División o Módulo solo que esta funcion es alimentada con el resultado de una funcion de hashing previa.

Se penso el hash para trabajar con registros de tamaño variable que se almacenaran en cubetas con capacidad para varios registros.

### Cubeta:

Overflow (1 byte)	FreeSpace (2 bytes)	Count (2 bytes)	bytes
-------------------	---------------------	-----------------	-------

El tamaño de las cubetas es parametrizable como tambien el tipo de dato que se almacena en la tabla. De acuerdo al tipo de registro a insertar se tomaron distintos multiplos de 512 para determinar el tamaño de la cubeta.

Para determinar la cantidad optima de cubetas en diferentes escenarios se consideraron diferentes parametros:

C: numero de registros por cubeta

R: numero total de registros almacenados

DE: Densidad de empaquetamiento la cual se fijo en 0.7

C y R son parametros estimativos que deben determinarse conociendo los registros a insertar.

Asi se despeja el numero optimo de cubetas D, como  $D = R/(C*DE)$ .

Finalmente se verifica que D sea primo y en caso contrario se toma el primo mas cercano.

### Primitivas implementadas:

Se implementaron las siguientes primitivas:

- Creación: Se crea un nuevo archivo y se calcula la cantidad de buckets necesarias.
- Carga: Se abre un archivo y se determina la cantidad de buckets existentes.

- Inserción: Se intenta insertar un registro y lanza una excepción en caso de que ya exista.
- Recuperación: Se intenta recuperar un registro y lanza una excepción en caso de que no exista.
- Actualización: Se intenta actualizar un registro y lanza una excepción en caso de que no exista.
- Borrado: Se intenta borrar un registro y lanza una excepción en caso de que no exista.

## Archivos:

Se utiliza un solo archivo que es el que contiene la totalidad de las cubetas.

## Diagrama de clases

Se presentan las

