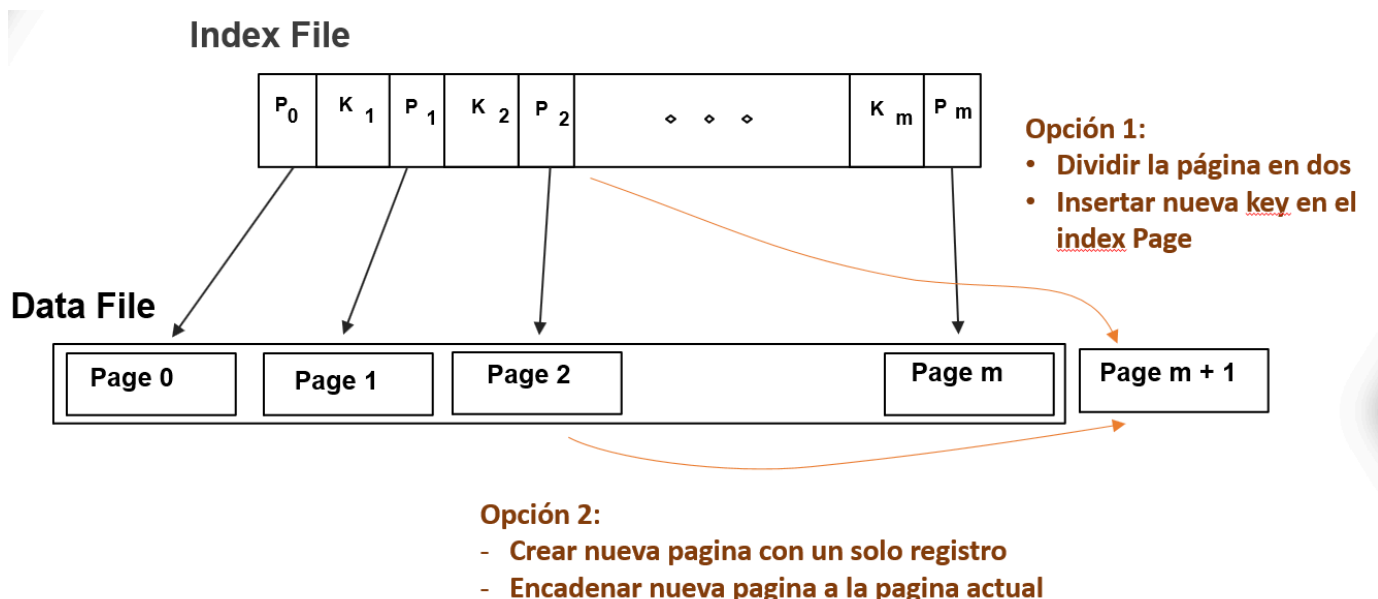


## Laboratorio 3: ISAM (Sparse Index) & Static Hashing

### P1 (12 pts). ISAM (Sparse Index)

Implementar ISAM con un solo nivel de indexación (sparse index).

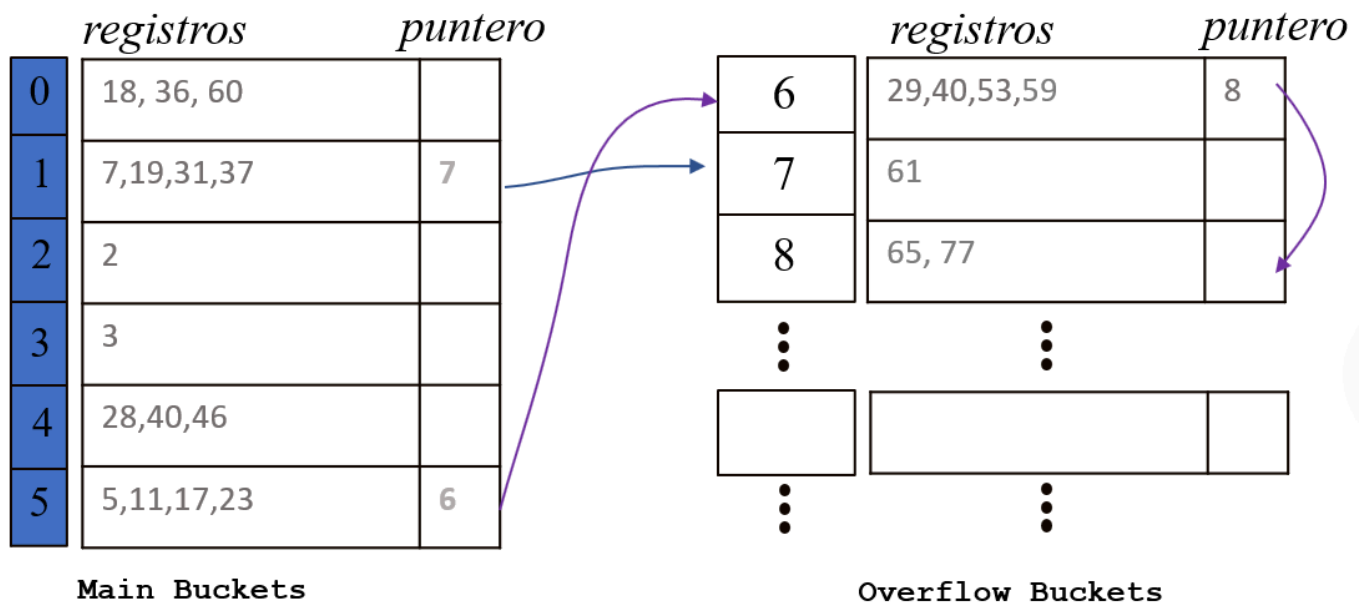
1. Diseñar la estructura de página de datos y de índice. El factor de bloque y la cantidad de entradas varían según el tamaño del buffer, puede mantener dichos valores como constantes globales.
2. Construir el archivo de datos con registros iniciales ordenados y organizados en páginas. El índice apunta al primer registro de cada página.
3. Inserción: Si la página está llena, dividirla y ubicar la nueva página al final del archivo. Actualizar punteros.
4. Búsqueda: Usar el índice para localizar la página y buscar el registro.
5. Eliminación: Eliminar el registro. ¿Qué hacer si una página queda vacía?



### P2 (8 pts). Static Hashing

Implementar Static Hashing con buckets y desbordamiento encadenado.

1. Diseñar la estructura del bucket y la clase que gestiona el archivo.
2. Inserción: Si el bucket está lleno, crear y enlazar un bucket overflow.
3. Búsqueda: Localizar el bucket y buscar el registro, incluyendo buckets overflow.
4. Eliminación: Eliminar el registro. ¿Qué hacer si un bucket queda vacío?



## Entregable

- Código fuente en Python con las implementaciones de ambas técnicas.
- Pruebas Funcionales debe incluir pruebas funcionales de cada uno de los métodos.
- Analisis en el informe incluir el análisis de las ventajas y desventajas de dichas técnicas.