**Archivos de acceso secuencial indizado (ISAM)**

En términos generales la exploración secuencial de los archivos con el objeto de localizar un registro es un proceso q consume tiempo excesivo pero se le emplea a menudo para localizar con precisión el registro buscado dentro de un área reducida.

El diseño de archivo secuencial indizado intenta superar el problema de acceso inherente a la organización de archivos secuencial sin perder todos los beneficios de los archivos secuenciales. Un archivo secuencial indizado consta de tres cosas en el disco:

1. Un índice del archivo: El área de índice es un archivo secuencial q contiene las claves del primer registro de cada bloque físico del archivo y la dirección de acceso al primer registro del bloque.
2. Un área principal: Contiene los registros de datos clasificados en orden ascendente por el campo clave.
3. Un área de desborde: Contiene los registros q no se pueden situar en el área principal.

Los archivos secuenciales indizados están diseñados para aplicaciones q requieren lo siguiente:

* Procesamiento secuencial de todo el archivo.
* Acceso directo a registros por separado.

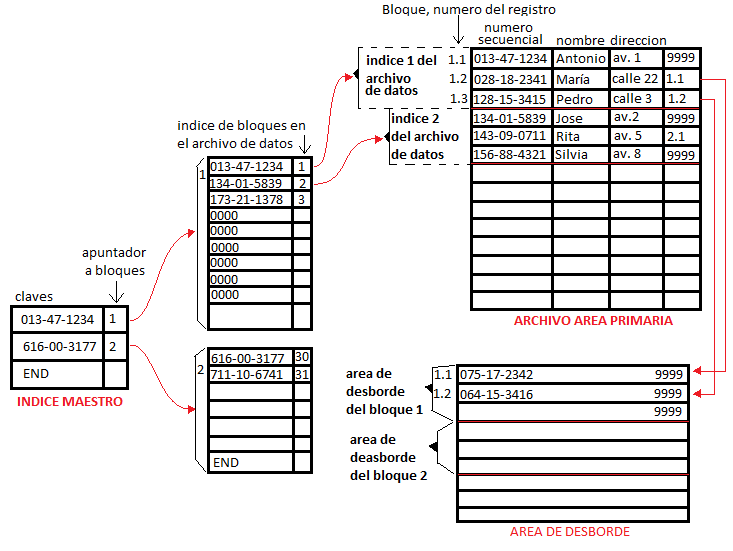
Los archivos secuenciales indizados permiten dos tipos de procesamiento:

**Procesamiento secuencial:** En el que los registros se piden precisamente en el orden en el que están distribuidos en el archivo.

**Procesamiento aleatorio:** En el que se llega al registro en un orden cualquiera, sin nada q ver con su distribución física.

La importancia relativa de estas dos formas de procesamiento influye sobre la elección de la organización secuencial indizada para ser utilizada en una aplicación dada.

**Ejemplo de archivo secuencial indizado:**

****

En este ejemplo observamos todos los elementos que conforman una estructura de archivos secuencial indizado clásico.

Un diccionario es un archivo de acceso secuencial indizado. Cada página es un bloque que posee dos niveles de índices o indización, el nivel superior es la letra inicial de la palabra y el nivel inferior las cabeceras de cada página.

**Estructura de los archivos secuenciales indizados**

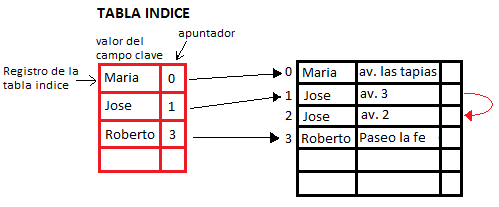
**Índice:** Un índice consiste en un conjunto de registros almacenados en una tabla, uno para cada bloque de registro de dato. El registro de la tabla de índice está formado, por un atributo clave para ese registro y un apuntador hacia el registro dato en el archivo. El índice permite un acceso inmediato a un registro por medio de un campo clave, debido a que el índice conserva un orden de acuerdo con un atributo clave.

**Tipos de índices: (Densos y No Densos)**

**Índices Densos**

Existe un registro índice para cada valor distinto de la llave de acceso en el archivo. El registro de índice contiene el valor de clave de acceso y un apuntador al registro.

**Ejemplo:**

****

Observe que en el caso de registro repetido con el mismo valor para el campo clave no existen múltiples entradas para la tabla índice. Con una sola entrada basta.

También observe que para los índices densos existe una relación uno a uno con el archivo de dato.

**Índices No Densos o Anclas de Bloques**

En este caso la tabla índice contiene solamente el primer registro de cada bloque de registros del archivo de registros.  
  
La ventaja de un índice no denso o ancla de bloque radica en que permite el acceso rápido a un bloque del archivo. Los registros individuales en un bloque pueden encontrarse mediante una búsqueda dentro del bloque de manera que no es necesario conservar en el índice una anotación para cada registro sino una referencia a un registro por bloque. Al registro que corresponde la referencia se le denomina Punto ANCLA y en el índice se conservan solamente el valor llave del ancla y el apuntador al bloque.

El costo de buscar un registro dentro de un bloque de archivo es mínimo, ya que todo el bloque se trae a memoria siempre que se necesite y pueda conservarse en un buffer. El número de entradas o registros en un índice con ancla de bloques es:

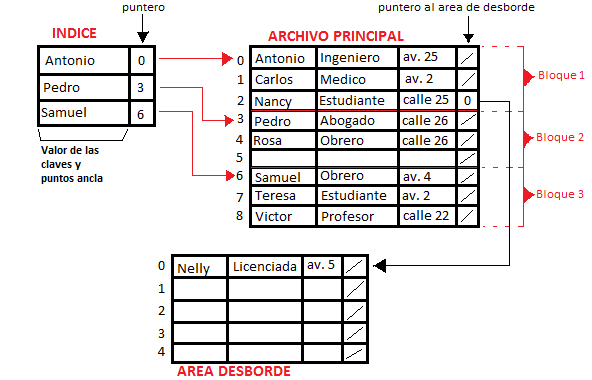
Número de registros esperados en el archivo/ Tamaño del buffer

**NEI = N/BUFFER**

El tamaño del registro almacenado en el índice es igual al tamaño de la clave más el tamaño del puntero.

**TRAI = Tc + Tp**

**Ejemplo de índice no denso o ancla de bloque**

****

**Para localizar un registro**  en un índice no denso. Primero se busca en el archivo Indice el registro índice que contenga el valor de la llave de acceso más grande pero menor igual que el valor de la llave de acceso que se está buscando.

Una vez encontrado el registro índice que cumpla con esa condición nos posicionamos en el bloque que apunta este registro en el archivo principal, la lectura del archivo principal se realiza en forma secuencial hasta que se encuentre el registro solicitado o se encuentre un apuntador al área de desborde. En el área de desborde se realiza una búsqueda secuencial hasta que se encuentre el registro o un apuntador nulo.

**Clases de índices**

* Índices primarios **(índice no denso)**
* Índices secundarios **(índice denso)**

**Índices primarios**

El índice primario para un archivo secuencial indizado es aquel que se construye con el mismo campo que se ha utilizado para determinar la secuencia de los registros dentro del archivo de registro. La clave de búsqueda en un índice primario es normalmente (aunque no siempre) la clave primaria.

**Índices secundarios**

Los índices secundarios son aquellos que utilizan como campo clave de indización a un campo que no determina el orden de la secuencia de los registros en el archivo de datos. Ejemplo de campo clave secundaria = dirección.

**Razón de abanico de salida de un índice**

Un parámetro importante de un índice es la capacidad de hacer referencia clave a un bloque del índice al archivo de datos, es decir su abanico de salida. La amplitud de salida **Y** es el cociente del tamaño del bloque **B** y el espacio ocupado por cada registro índice **Tc + Tp**.

**Cantidad de niveles de un índice**

**CNI = ( ln(NEI/B) / ln(y) )**

Usualmente esto da un número que varía de 1 a 3, valores mayores a estos ocurren según el tamaño del archivo clave.

Para evaluar el número de niveles del índice que podrían necesitarse, se tomara el ejemplo de un archivo bastante grande (un millón de registros) y un índice no denso o ancla de bloque. A fin de calcular el tiempo de acceso para recuperar un registro, es necesario saber cuántos niveles de índice se requerirán para un archivo de este tamaño.

**Diseño del índice independiente del hardware**

Si conocemos el tamaño del bloque **B** es igual a 2000 bytes, con un tamaño de clave Tc = 16 bytes, un tamaño de apuntador Tp = 4 bytes y un tamaño de registro Tr = 200 bytes. El factor de bloque es fácil de calcular:

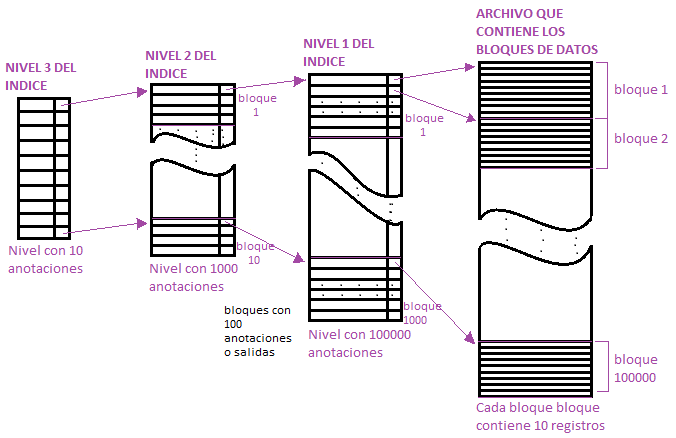
fb = 10 registros por bloque, por lo tanto para almacenar el millón de registros se requieren 100000 bloques de datos, por lo tanto, el mismo número de anotaciones del apuntador índice. En este caso el tamaño de una anotación del índice es Tc + Tp = 20 bytes.

Ahora bien la ecuación de razón de salida del índice , de manera que las 100000 anotaciones del nivel más bajo del índice ocupan anotaciones del índice del segundo nivel. Este índice de segundo nivel ocupara un total de 20x1000 = 20000 bytes.

Esta última cifra sigue siendo excesiva para almacenamiento en el núcleo de manera que se creara un tercer nivel de índice en el que se requerirán solamente anotaciones en el índice de tercer nivel ocupando solo 200 bytes.  
  
El término “índice maestro” se refiere al nivel superior de los índices.

Los niveles de los índices se enumeran desde el 1 para el nivel más cercano a los datos hasta x (en este caso 3) para el último nivel o índice maestro.

**Ejemplo en forma grafica**



Durante el procesamiento es común para la reducción de costo de acceso que el índice maestro (nivel x) permanezca disponible en la M.P. una vez que se han leído los niveles (x-1,… ,1) del índice se utilizan a partir del disco. Con este esquema se requieren un total de x accesos a los bloques del disco para recuperar el registro de datos del archivo principal.

El índice maestro se almacena en memoria principal cuando se está trabajando con el índice. Cuando no, el índice maestro está almacenado en memoria secundaria.

**Diseño del índice orientado al hardware**

Usando un disco con cilindros de 19 pistas capaces cada uno de contener 14mil bytes y B=2mil se encuentran en cada cilindro 19**x** 14mil**/**2mil **=** 133 bloques. Las anotaciones del índice que coinciden con unidades secuenciales del hardware no necesitan un campo apuntador ya que la anotación 1 corresponde al bloque 1, la anotación 2 al bloque 2 y así sucesivamente. El índice para los datos contenidos en el nivel 1 requerirá una notación por bloque cuando más 133xTc = 1862 bytes o un bloque en cada cilindro dejando 132 bloques para datos.  
  
También habrá un índice de segundo nivel para los índices de cilindro; en esta organización su tamaño depende del número necesario para este archivo. El archivo de dato ocupa n/factor de bloques, donde n es la cantidad de registros a insertar 1000000/10 = bloques o /132=758 cilindros. El índice para cilindros tampoco requiere un campo apuntador, ya q simplemente tiene una anotación por cilindro secuencial. En la práctica el emplear índices de primer nivel residente en el cilindro, el índice de segundo nivel residirá dentro de cada paquete de disco dado.

**Para el siguiente ejemplo**

Supuestos bajo los cuales se hacen los algoritmos de búsqueda:

Utilizamos el último registro para dos cosas:

1. Saber cuántos registros tiene el bloque.
2. Insertar ahí cuando el bloque esta full.

