

GESTIÓN Y ADMINISTRACIÓN DE SISTEMAS GESTORES Y BASES DE DATOS

Jesús Pedro Romero Castellano



educàlia
EDITORIAL

GESTIÓN Y ADMINISTRACIÓN DE SISTEMAS GESTORES Y BASES DE DATOS

Jesús Pedro Romero Castellano

Segunda edición, 2014

Autor: Jesús Pedro Romero Castellano

Maquetación: Daniela Vasilache

Edita: Educàlia Editorial, S.L.

Imprime: Igràfic

ISBN: 978-84-15161-63-9

Depòsit Legal: V850-2013

Printed in Spain/Impreso en España.

Todos los derechos reservados. No está permitida la reimpresión de ninguna parte de este libro, ni de imágenes ni de texto, ni tampoco su reproducción, ni utilización, en cualquier forma o por cualquier medio, bien sea electrónico, mecánico o de otro modo, tanto conocida como los que puedan inventarse, incluyendo el fotocopiado o grabación, ni está permitido almacenarlo en un sistema de información y recuperación, sin el permiso anticipado y por escrito del editor.

Alguna de las imágenes que incluye este libro son reproducciones que se han realizado acogiéndose al derecho de cita que aparece en el artículo 32 de la Ley 22/18987, del 11 de noviembre, de la Propiedad intelectual. Educàlia Editorial agradece a todas las instituciones, tanto públicas como privadas, citadas en estas páginas, su colaboración y pide disculpas por la posible omisión involuntaria de algunas de ellas.

Educàlia Editorial, S.L.

Ayda de les Jacarandes 2 loft 327 46100 Burjassot-València

Tel. 960 624 309 - 963 768 542 - 610 900 111

E-Mail: educaliaeditorial@e-ducalia.com

Capítulo 1. Sistemas de ficheros y bases de datos

- 1.1 Sistema de ficheros
- 1.2 Clasificación de los ficheros según su función
- 1.3 Sistemas de bases de datos
- 1.4 Tipos de sistemas gestores de bases de datos
- 1.5 Bases de datos distribuidas (BDD)
- 1.6 Bases de datos cliente-servidor

Capítulo 2. Instalación y uso de MySQL

- 2.1 ¿Qué es MySQL?
- 2.2 Arquitectura típica de MySQL
- 2.3 Obtención e instalación de MySQL
- 2.4 Prácticas desarrolladas

Capítulo 3. SQL3 y MySQL

- 3.1 El lenguaje SQL3
- 3.2 Lenguaje de definición de datos (LDD o SQL estático)
- 3.3 Lenguaje de manipulación de datos (LMD o SQL dinámico)
- 3.4 Funciones de resumen
- 3.5 Predicados
- 3.6 Algunas funciones de MySQL
- 3.7 Ejercicios de SQL resueltos
- 3.8 Ejercicios propuestos

Capítulo 4. SQL-PSM y triggers

- 4.1 SQL/PSM (Persistent Stored Modules, ISO/IEC 9075-4:2003)
- 4.2 Tipos de instrucciones en PSM
- 4.3 Variables en procedimientos almacenados
- 4.4 Consultas (SELECT) en procedimientos almacenados
- 4.5 Cursores
- 4.6 Modificación, eliminación y consulta de procedimientos almacenados
- 4.7 Condiciones y manejadores de error (handlers)
- 4.8 Disparadores en SQL3 (TRIGGERS, SQL-ISO/IEC 9075-2:1999)
- 4.9 Triggers en MySQL
- 4.10 Relación de ejercicios de PSM y triggers

Capítulo 5. Utilización de MySQL Workbench para consultas y edición de datos

- 5.1 ¿Qué es MySQL Workbench?
- 5.2 Realización de consultas
- 5.3 Edición e inserción de datos

Capítulo 6. El modelo de entidad-relación extendido (ERE)

- 6.1 Introducción
 - 6.2 Elementos del modelo Entidad/Interrelación
 - 6.3 Método de diseño
 - 6.4 Ejercicios propuestos de modelado EER
- ## **Capítulo 7. Diseño y modelo lógico de datos relacional**
- 7.1 Introducción
 - 7.2 Estructura lógica de los datos en el modelo relacional
 - 7.3 Reglas de integridad del modelo relacional
 - 7.4 Manipulación de datos en el modelo relacional
 - 7.5 Normalización de las relaciones

Capítulo 8. Utilización de MySQL Workbench para el modelo ER extendido

- 8.1 ¿Qué es MySQL Workbench?
- 8.2 Descarga e instalación de MySQL Workbench
- 8.3 Simbología usada por MySQL Workbench
- 8.4 Entidades y atributos de entidades
- 8.5 Interrelaciones M:N
- 8.6 Cardinalidades máximas diferentes de n
- 8.7 Cardinalidades mínimas mayores que 0 y 1
- 8.8 Interrelaciones 1:1
- 8.9 Restricciones sobre las interrelaciones

- 8.10 Subtipos y supertipos
- 8.11 Agregaciones
- 8.12 Generación de código SQL

Capítulo 9. Administración de bases de datos

- 9.1 Administración de de datos y administración de bases de datos
- 9.2 Diseño de la base de datos
- 9.3 Creación de las estructuras de almacenamiento
- 9.4 Arquitectura del SGBD
- 9.5 Seguridad
- 9.6 Integridad
- 9.10 Auditoría
- 9.11 Copias de seguridad
- 9.12 Supervisar el rendimiento y responder a cambios en los requisitos
- 9.13 Instalar y actualizar el software de la base de datos

Capítulo 10. Administración de MySQL

- 10.1 Arquitectura y componentes de MySQL
- 10.2 Privilegios en MySQL
- 10.3 Tablas de permisos
- 10.4 Creación de usuarios y concesión de privilegios
- 10.5 Mostrar los privilegios de una cuenta
- 10.6 Eliminar privilegios y usuarios
- 10.7 Cambiar y restablecer contraseñas
- 10.8 Copias de seguridad
- 10.9 Recuperar copias de seguridad
- 10.10 Administración de MySQL con herramientas de interfaz gráfica de usuario
- 10.11 Ejercicios propuestos

Capítulo 11. Instalación de Oracle 10g

- 11.1 Instalación de Oracle Database Server 10g Release 1 (10.1.0.2.0) para Windows

Capítulo 12. Administración de Oracle 10g. Herramientas gráficas

- 12.1 Componentes principales de la arquitectura del servidor Oracle
- 12.2 Conceptos sobre seguridad en Oracle 10g
- 12.3 Administración de la seguridad en Oracle 10g
- 12.4 Creación de una base de datos con Database Configuration Assistant
- 12.5 Ejercicios propuestos 305

Capítulo 13. Programación de páginas web de acceso a datos con PHP

- 13.1 El protocolo http
- 13.2 CGI (Common Gateway Interface)
- 13.3 Lenguajes scripts
- 13.4 Conectividad Abierta de Bases de Datos (ODBC)
- 13.5 Arquitectura de ODBC
- 13.6 Programando con ODBC: generalidades y ejemplos
- 13.7 Funciones ODCB en PHP
- 13.8 Ejercicios
- 13.9 Soluciones de los ejercicios

Capítulo 14. Proyecto integrador

- 14.1 Proyecto de construcción de una aplicación de gestión y visita de museos

“Gestión y administración de sistemas gestores y bases de datos”

Tres han sido los objetivos fundamentales que han estado siempre presentes a la hora de elaborar este libro.

En primer lugar que constituya la principal herramienta de trabajo del profesor para impartir las clases siendo la prolongación y desarrollo de su programación didáctica.

En segundo lugar y no menos importante, que los contenidos estén actualizados y tengan un adecuado equilibrio entre conceptos y práctica, permitiendo al alumno alcanzar las capacidades terminales o competencias de los módulos, haciéndole ver de forma rápida la aplicación de lo aprendido.

En tercer lugar se ha tenido en cuenta la fácil disponibilidad y nulo coste que supone el uso de herramientas de software libre o de licencia GPL; además de la oportunidad que ofrecen dichas herramientas al alumno para el autoempleo.

Por este motivo todo el software usado para desarrollar los contenidos es de este tipo, con calidad y uso contrastados como MySQL, Oracle, MySQL Workbench, Apache, etc.

En resumen el libro es el resultado de la experiencia acumulada de varios años de docencia en centros públicos en los módulos relacionados con las bases de datos de los ciclos superiores de Informática. Cada capítulo se concibe como el desarrollo de una unidad didáctica, equilibrado entre conceptos y práctica y diseñado para que contribuya a la adquisición de las correspondientes capacidades terminales teniendo en cuenta la escasez de medios económicos y la oportunidad para el autoempleo.

El libro se recomienda usarlo como libro de texto en los módulos de Sistemas gestores de bases de datos del ciclo superior de Administración de Sistemas Informáticos y Gestión de bases de datos del ciclo de Administración de sistemas informáticos y en red ya que cubre los contenidos de dichos módulos.

También se puede usar en el módulo Administración de sistemas gestores de bases de datos de éste último ciclo.

Los contenidos se organizan en tres bloques con sus correspondientes temas o capítulos y un proyecto integrador final práctico. A continuación se detallan estos bloques.

BLOQUE 1º: GESTIÓN DE SISTEMAS GESTORES DE BASES DE DATOS

Dedicado principalmente al diseño, creación de bases de datos relacionales y estudio del lenguaje SQL3 usando como sistema gestor de bases de datos MySQL y como herramientas gráficas de diseño consulta y edición MySQL Workbench . Se compone de los siguientes temas:

TEMA 1.- SISTEMAS DE FICHEROS Y BASES DE DATOS

TEMA 2.- INSTALACIÓN Y USO DE MySQL

TEMA 3.- SQL3 Y MySQL

TEMA 4.- SQL-PSM Y TRIGGERS

TEMA 5.- UTILIZACIÓN DE MySQL Workbench PARA COSULTAS Y EDICIÓN

TEMA 6.- MODELO ER EXTENDIDO (EER)

TEMA 7.- DISEÑO Y MODELO LÓGICO DE DATOS RELACIONAL

TEMA 8.- UTILIZACIÓN DE MySQL Workbench PARA EL MODELO EER

BLOQUE 2º: ADMINISTRACIÓN DE SISTEMAS GESTORES DE BASES DE DATOS

Dedicado a la administración de las bases de datos en los aspectos de seguridad e integridad usando las herramientas de los sistemas gestores MySQL y Oracle 10g. Se compone de los siguientes temas:

TEMA 9.- ADMINISTRACIÓN DE BASES DE DATOS

TEMA 10.- ADMINISTRACIÓN DE MySQL.HERRAMIENTAS GRÁFICAS

TEMA 11.- INSTALACIÓN DE ORACLE 10g

TEMA 12.- ADMINISTRACIÓN DE ORACLE 10g. HERRAMIENTAS GRÁFICAS

BLOQUE 3º: PROGRAMACIÓN WEB CON ACCESO A BESE DE DATOS

Dedicado a la construcción de páginas web de acceso a bases de datos usando PHP. Este bloque está relacionado con los módulos de implantación de aplicaciones web y lenguajes de marcas y sistemas de gestión de información ya que se requiere un servidor web confi gurado y funcionando y conocimientos de HTML. Se compone del siguiente tema:

TEMA 13.- PROGRAMACIÓN DE PÁGINAS WEB DE ACCESO A DATOS CON PHP

BLOQUE 4º: PROYECTO INTEGRADOR

Este bloque es totalmente práctico y su fi nalidad es fomentar la capacidad de síntesis, el trabajo en equipo y por objetivos, la organización, coordinación y elaboración de documentación. En concreto se trata de la especi fi cación de los requisitos de un proyecto a realizar por el alumno con el fi n de diseñar, crear, administrar una base de datos y diseñar una aplicación o interfaz web de la base de datos que permita operaciones de consulta, actualización, inserción y eliminación de información a determinados usuarios de la base de datos.

Capítulo 1.

Sistemas de ficheros y bases de datos

Sistema de ficheros

Un **sistema de ficheros** es un conjunto de programas creados y mantenidos por un programador que prestan servicio a los usuarios y cada programa maneja sus propios datos contenidos en ficheros.

Los sistemas de ficheros surgieron al tratar de informatizar el manejo de los archivadores manuales de las empresas con objeto de proporcionar un acceso más eficiente a los datos. Esto supone un modelo descentralizado en el que cada departamento de la empresa posee sus propios ficheros con datos y formatos posiblemente incompatibles y con diferentes formas de manejar los ficheros.

Conceptos sobre ficheros

Fichero o archivo: conjunto de registros lógicos del mismo tipo.

Registro lógico: unidad de información compuesta por campos que se refieren a un objeto, concepto o entidad.

Los registros lógicos pueden ser:

- De longitud fija si todos los campos tienen igual tamaño.
- De longitud variable si hay campos de distinta longitud.
- De longitud indefinida.

Registro físico: unidad de información que se transfiere en una operación de E/S entre el soporte de almacenamiento y la memoria principal, también se le denomina **bloque o cluster**, y puede ser inferior, superior o igual al tamaño del registro lógico. En la práctica un registro físico siempre o casi siempre contiene varios registros lógicos y tiene un tamaño fijo.

Campo: es el grupo de datos nominado, es decir que tiene un nombre, más pequeño y con significado.

EJEMPLOS

1.- Definición y declaración de un registro lógico en pseudo código:

```
tipo DAT_PERS es registro compuesto de
    NOMBRE      es alfanumérico
    DIRECCIÓN  es alfanumérico
    EDAD       es entero
    DNI        es entero
```

Fin registro

```
DATOS_PERSONALES es de tipo DAT_PERS
```

NOMBRE, DIRECCIÓN y EDAD son los campos del registro DATOS PERSONALES

2.- Definición y declaración del mismo registro en C:

```
struct  DatPers
    char      nombre [50];
    char      direccion[50];
    int       edad;
    long int  dni;
} DatosPersonales;
```

Clave de un registro: es el campo o combinación de campos del registro que identifica de forma unívoca al objeto o entidad representada por el registro. Esto significa que no puede haber dos entidades con esa misma clave. Un registro puede tener más de una clave (clave primaria y claves alternativas).

EJEMPLO

En el registro `DatosPersonales` el campo clave debería ser `dni`.

Operaciones más frecuentes sobre ficheros

- **Creación**
Consiste en grabar los registros del archivo en un soporte de almacenamiento según una determinada organización. Los registros se pueden grabar a través del teclado, desde otro fichero o bien fusionando varios archivos en uno.
- **Apertura y cierre**
Cuando se abre un archivo el sistema operativo, de forma transparente para el usuario, reserva estructuras de memoria intermedia (buffers) y las asocia al archivo físico en su soporte. Cuando una aplicación hace una lectura o escritura del archivo en realidad lo hace en estas estructuras de memoria. Cuando se cierra el archivo los datos que contienen esos buffers se escriben en el soporte y se liberan las estructuras de memoria.
- **Consulta**
Permite buscar y leer registros concretos dentro de un fichero a través de su clave o identificativo.
- **Actualización**
Consiste en dar de alta, baja o modificar los registros del archivo.
- **Listado**
Consiste en volcar el contenido del fichero por pantalla o impresora.
- **Reorganización**
Consiste en eliminar físicamente del archivo los registros dados de baja, de forma que se liberen los huecos intermedios existentes juntando todos los huecos por un lado y los registros por otro mejorando así la rapidez de acceso.

Práctica

Comprueba cuál es el tamaño en KB del clúster en tu sistema de archivos

Clasificación de los ficheros según su función

Archivo permanente o maestro

Es aquel cuyo contenido ha de retenerse durante un periodo más o menos largo de tiempo. Dependiendo de las actualizaciones que se realicen se distinguen los siguientes archivos maestros:

- *Archivos de constantes*
Sus registros se modifican con muy poca frecuencia. Contienen información principalmente para ser consultada.
EJEMPLO
El fichero de personal de una empresa conteniendo los datos de sus empleados: nombre, dirección, etc.
- *Archivos de situación*
Contiene información que se actualiza al día o en periodos cortos de tiempo. Nos dan información acerca de la situación o estado de algo en ese periodo de tiempo.
EJEMPLO
El archivo que guarda los movimientos de una cuenta corriente.
- *Archivos históricos*
Sus registros no se actualizan nunca. Lo único que se hace es añadir más registros a este tipo de archivos de forma que todo el archivo refleje una historia.

Archivo de transacciones o de movimientos

Cada registro de este archivo es una transacción e indica que operación, añadir, modificar o borrar, hay que realizar sobre un archivo maestro. Una vez hecho esto el archivo de movimientos se borra o destruye aunque también es común que se guarde durante cierto tiempo por si hay que reconstruir el maestro.

EJEMPLO

El fichero de movimientos con las altas, bajas y modificaciones de los artículos de un almacén.

Archivo de maniobra, de trabajo, intermedio o temporal

Son ficheros que contienen información de los resultados intermedios de un proceso o programa. Sirven como datos de entrada a otro proceso y se suelen eliminar al final del proceso.

Organización de los ficheros

Organización de un fichero es lo mismo que orden físico. Es la forma de almacenar o colocar los registros lógicos del fichero en el soporte de información (disco duro, cinta, disquete, CDROM, etc.). El orden de las posiciones físicas en que se colocan los registros no tiene porque corresponder con el orden lógico de sus claves. El orden físico determina la organización del fichero.

Existen dos organizaciones fundamentales a partir de las cuales se construyen todas las demás. Estas dos organizaciones son la **secuencial** y la **aleatoria**, **directa** o **relativa**.

Organización secuencial

En este tipo de organización los registros se almacenan en posiciones físicas contiguas dentro del soporte siguiendo el orden de sus claves, es decir el registro con la primera clave en el orden lógico estaría en la primera posición, el registro con la segunda clave en la segunda posición y así sucesivamente.

REGISTRO 1	REGISTRO 2	REGISTRO 3	...
------------	------------	------------	-----

Ventajas de esta organización:

- ✓ rápido acceso a un registro colocado a continuación de otro

Desventajas de esta organización:

- ✓ para acceder a un registro concreto hay que pasar previamente por todos los anteriores a él, es decir acceso lento en general a cualquier registro.
- ✓ es necesario volver a crear el archivo si se desea borrar, añadir o modificar un registro con objeto de que se mantenga el orden.

Variantes de la organización secuencial

Los inconvenientes de la organización secuencial pura vista antes pueden obviarse usando los **punteros** y los **índices**.

Puntero :

Variable que contiene la dirección o posición de memoria de otra variable. Se dice que apunta a otra variable.



Índice :

Tabla que contiene registros con pares de valores **(cla,pos)**, donde:

cla es el valor del campo clave de un registro de un archivo y **pos** la posición del registro en el archivo.

Si **pos** es la posición de un registro individual del archivo con el mismo valor de su campo clave que **cla** entonces tendremos un **índice denso** (el índice apunta al registro y habrá tantos registros en el índice como el archivo principal).

Si **pos** es la posición en el archivo de un grupo de registros (a este grupo se le denomina **página**) entonces tendremos un **índice no denso** (el índice apunta a una página que contiene el grupo de registros)

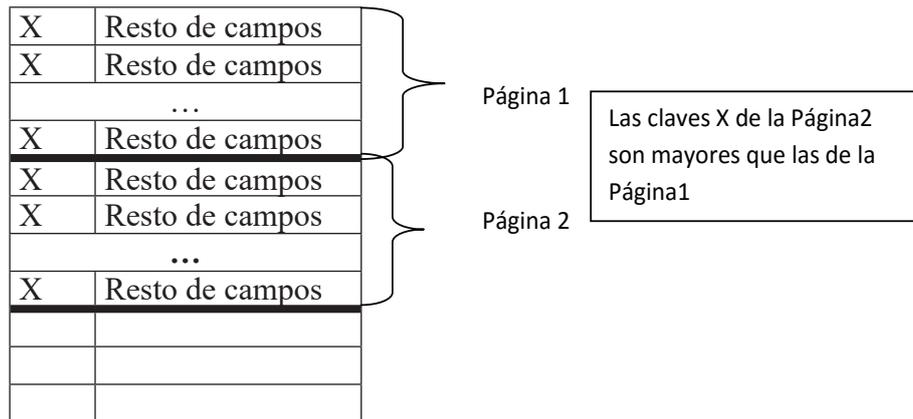
Organización secuencial indexada (ISAM=Indexed Sequential Access Method)

Se distinguen tres áreas en esta organización:

- **Área de datos o fichero de datos**

Contiene los registros colocados secuencialmente por orden de llegada (inserción) dentro de páginas. El valor máximo de las claves de los registros de una página i , que denotaremos C_i , junto con la posición de comienzo de la página (P_i) se coloca en la tabla de índices. Si representamos por X el valor de las claves de los registros en las páginas, entonces una página i contiene el siguiente conjunto de claves de registros:

$\{X/ C_{i-1} < X \leq C_i\}$, es decir todas las claves de la página i son menores o iguales que C_i y mayores que las claves de la página $i-1$



• **Área o tabla de índices:**

Es un índice no denso en el que cada registro apunta a una página.

$$TABLA\ INDICES = \{(C_i, P_i)\} \quad i=1 \dots n$$

donde:- C_i es la máxima clave de los registros que están en la página i , es decir la página i contiene registros cuya clave es menor o igual que C_i .

- P_i es la posición física o de comienzo de la página i

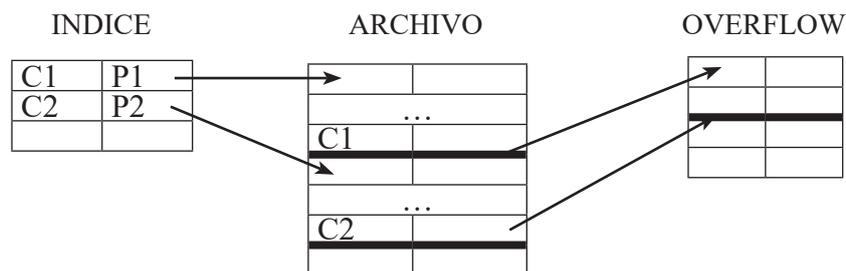
- $C_i \leq C_{i+1}$, es decir los registros del índice están ordenados

C1	P1
C2	P2
C3	P3
...	

Tabla de índices

• **Área de desbordamiento u overflow**

Cuando un registro no puede colocarse en la página que le corresponde según el valor de su clave porque dicha página está llena, entonces se utiliza la página de desbordamiento correspondiente y se busca un hueco para colocar el registro.



-Organización secuencial indexada-

Búsqueda de un registro

La búsqueda de un registro con clave C supone:

1. Buscar en el índice la primera clave C_i tal que $C \leq C_i$
2. Acceder directamente a la página P_i
3. Buscar secuencialmente el registro en la página P_i y si no está buscar secuencialmente en la página de overflow correspondiente. Si no está en la página de overflow entonces ese registro no existe en el archivo.

Introducción de un registro

La introducción de un registro con clave C en el archivo supone

1. Buscar en el índice la primera clave C_i tal que $C \leq C_i$
2. Acceder directamente a la página P_i
3. Buscar secuencialmente el primer hueco en la página P_i y si no hay hueco buscarlo secuencialmente en la página de overflow correspondiente. Si no hay hueco en la página de overflow entonces hay que reorganizar la estructura creando un nuevo fichero con páginas más grandes y pasar cada página de overflow a este fichero dejando las páginas de overflow vacías.

Ventajas de esta organización

La tabla de índices se mantiene en memoria principal ordenada de forma que se puede realizar una búsqueda binaria sobre ella o bien sobre la tabla de índices se puede construir un B⁺-árbol acelerando en cualquier caso las búsquedas.

EJEMPLO EN LA PRÁCTICA

Muchos sistemas gestores de bases de datos utilizan esta organización de archivos para las tablas que componen las bases de datos. Por ejemplo en las primeras versiones de **MySQL** el motor de almacenamiento por defecto al crear una tabla era ISAM. En la actualidad el motor de almacenamiento por defecto es InnoDB.

Práctica

Localiza en internet qué empresa inventó la organización ISAM y para qué la usó.

Organización secuencial encadenada(lista encadenada)

Esta variante de la organización secuencial se basa en las listas enlazadas y se caracteriza por:

- Utilizar dispositivos o soportes direccionables.
- Los registros se almacenan secuencialmente sin corresponder el orden físico al lógico.
- Disponer de una zona de desbordamiento (overflow) para almacenar los nuevos registros o en su defecto almacenarlos en posiciones físicas consecutivas a partir del último registro almacenado
- Cada registro dispone de un puntero que indica la dirección o posición del siguiente registro en la secuencia lógica.
- El puntero del último registro contiene un código especial (EOF) para indicar el fin del fichero.

Al principio cuando se crea este tipo de archivos, añadiendo en ellos el primer grupo de registros dichos registros se colocan adyacentes con una organización secuencial pura.

Ventajas

- ✓ Se pueden incluir nuevos registros sin necesidad de hacer una copia del archivo original.
- ✓ Se puede disponer en el mismo archivo físico de varias secuencias lógicas simplemente añadiendo más campos tipo puntero al registro que compone el archivo.

En la ilustración 1 los primeros registros que se añadieron al archivo, al crearlo, fueron los que están colocados consecutivamente en las posiciones 1 a 6.

La inserción se hace igual que un nodo en una lista enlazada para mantener el orden, es decir hay que cambiar los punteros con cada inserción. Por ejemplo considérese que se inserta el registro con clave Huertas en la situación de partida de la figura anterior, entonces hay que hacer que González apunte a Huertas y Huertas apunte a Iturriaga.

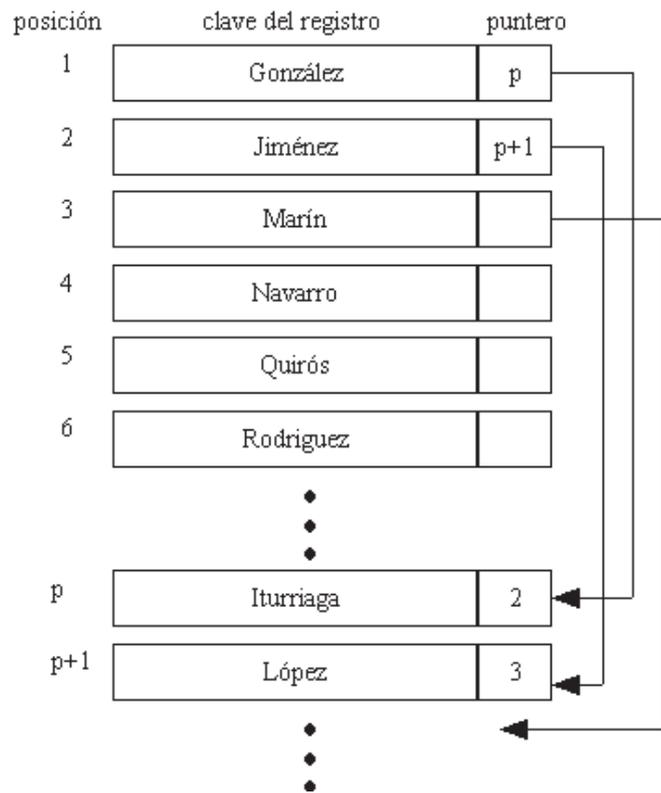


Ilustración 1. Estructura secuencial encadenada

Práctica

Localiza en internet qué es la FAT y si utiliza una organización secuencial encadenada.

Organización secuencial indexada encadenada

Esta organización es una combinación de la indexada y secuencial encadenada. Se dispone de índices y punteros.

En la organización secuencial indexada los registros ubicados en la zona de overflow quedan fuera de la secuencia lógica. En la organización secuencial indexada encadenada se usan los punteros para enlazar los registros de la zona de datos con la zona de overflow manteniendo así la secuencia entre ambos. Esto se ilustra en la siguiente figura.

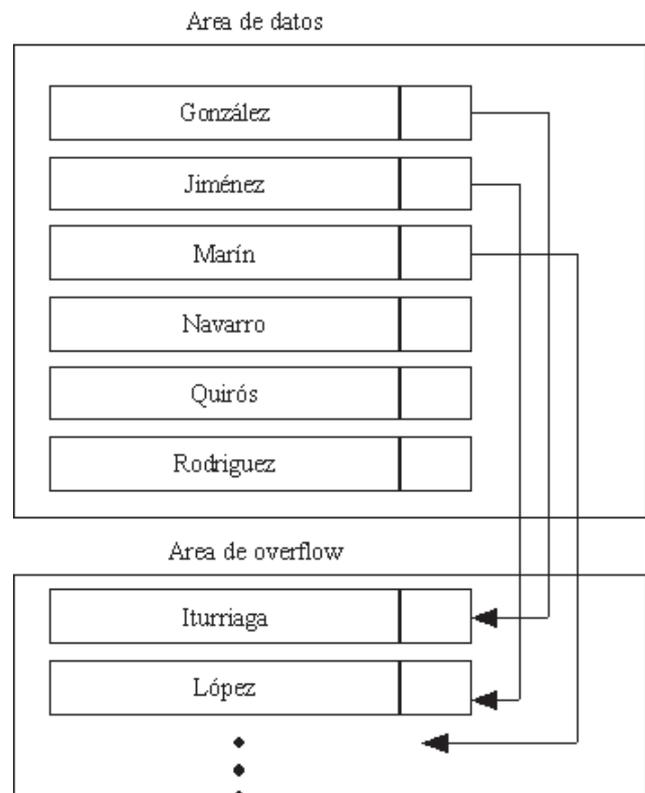


Ilustración 2. Organización secuencial indexada (falta tabla de índices)

Organización aleatoria , directa o relativa

Consiste en que la posición física donde se coloca el registro dentro del soporte se obtiene aplicando una transformación (denominada **función hash**) a la clave la cual da esa posición en el archivo. Es evidente que esta organización requiere soportes direccionables, es decir nunca podremos tener un archivo de acceso directo almacenado en una cinta ya que esta no es soporte direccionable.

La función hash asocia a una clave del espacio de claves de los registros una posición física en el archivo:

F: CLAVES → POSICIONES FÍSICAS

En la práctica la función hash implica crear un archivo vacío al principio y con un tamaño determinado, puesto que el rango de las posiciones ,la memoria disponible, será menor que el rango de las claves¹. Es decir hay que hacer una estimación inicial del tamaño que tendrá el archivo.

Algoritmos de transformación clave- dirección (funciones hash)

Estos algoritmos se pueden clasificar según dos métodos de direccionamiento:

- **Direccionamiento directo:** cuando el rango de las claves y de las direcciones o posiciones es el mismo.
- **Direccionamiento indirecto:** cuando el rango de claves es mayor que el de las direcciones con lo cual se producirán **sinónimos**.

Se produce un **sinónimo** cuando a dos o más registros con claves distintas la función hash les asigna posiciones iguales.

Se producen **huecos** cuando hay posiciones que no son ocupadas por ningún registro.

Utilización de la clave como dirección o método de direccionamiento directo

Es un direccionamiento directo. Requiere que la clave sea numérica y que el registro lógico sea de longitud fija. La función se aplica a la clave y se obtiene un número que indica la posición del registro.

Ejemplo

Un fichero de habitaciones de un hotel (con 200 habitaciones) con clave el número de habitación (suponiendo que al número de habitación no le sigue un dígito con el número de planta). Si el tamaño del fichero es de 200 registros una función hash para obtener la posición donde se grabaría la información de una habitación podría ser :

El resto de dividir el número de habitación entre 200

Es decir, la función hash viene dada por la siguiente expresión:

Posición buscada= num_ habitación mod 200

Así el registro de la habitación 100 se almacenaría en la posición 100 del fichero.

Cuestión:

¿Qué ocurre con esta función hash si por ejemplo el número de habitaciones del hotel es de 250 o de 400?

¹ Así fue en sus orígenes, aunque ahora con la gran cantidad de memoria disponible puede que esto no sea así.