

Ejercicios de Organización de Archivos

Organización de Registros y Procesamiento de archivos secuenciales

1. Hacer un programa que presente un menú cíclico con opciones para:
 - a. Crear un archivo binario de registros no ordenados de personas y cargarlo con datos almacenados en un archivo de texto, o abrir un archivo del mismo tipo existente.
 - b. Reportar en pantalla los datos de personas que cumplan años en un determinado mes o que tengan fecha de nacimiento 0.
 - c. Listar las personas en un archivo de texto.

Los nombres de archivos deben ser proporcionados por el usuario.

Los registros de persona deben contener la fecha de nacimiento, de tipo entero largo (las fechas se almacenan con formato aaaammdd), y el apellido y nombres en una cadena de 50 caracteres.

En el archivo de texto debe haber una línea por persona, primero la fecha de nacimiento y, separado por un espacio, el apellido y nombres.

2. Agregar al programa para crear y/o consultar el mes de cumpleaños de personas opciones para:
 - a. Añadir una o más personas al final del archivo con sus datos obtenidos por teclado.
 - b. Eliminar lógicamente una o más personas agregándole un asterisco al comienzo del campo para el apellido y nombres.
 - c. Ponerle o corregirle la fecha de nacimiento a una o más personas.

Las búsquedas deben ser por apellido y nombres.

Discutir el problema de las búsquedas y buscar una forma de que resulten efectivas.

3. Hacer una versión del programa previo que en lugar de realizar bajas lógicas las haga efectivas y realice las altas recuperando espacio libre. Discutir opciones para controlar los espacios libres.
4. Hacer una versión del programa anterior pero almacenando los registros con longitud variable en bloques de 64 bytes.
5. Una agencia de venta de autos y camiones con tres sucursales registra cada una de sus ventas en un archivo. Los datos de cada venta son: sucursal, tipo de automotor (auto o camión), tamaño, año y mes de venta y precio. Los autos pueden ser chicos, medianos o grandes; los camiones pueden ser medianos o grandes.

Hacer un programa **que recorriendo el archivo una única vez** reporte en un archivo de texto las estadísticas de ventas que se listan a continuación:

- a. Cantidad de autos vendidos por mes y por año
 - b. Cantidad de camiones vendidos por mes y por año
 - c. Ventas mensuales y anuales por sucursal y en toda la agencia
 - d. Tamaño de auto más vendido por mes y por año junto con las cantidades
 - e. Tamaño de camión más vendido por mes y por año junto con las cantidades
6. Dada la definición que sigue, defina la cabecera de una función y las estructuras de datos y variables en C para fusionar entre 2 y 15 archivos con registros de tipo Reg ordenados por el campo id, y explique cómo se controla el fin de los mismos en el proceso de fusión.

```
typedef struct { int id; ... } Reg;
```
7. Esquematizar la fase de ordenación para los registros representados por las claves que se listan abajo, empleando el método del montículo para la ordenación interna y el de selección y reemplazo para la externa, con buffers de lectura y de escritura con capacidad para dos (2) registros y buffer de ordenación con capacidad para cinco (5) registros:

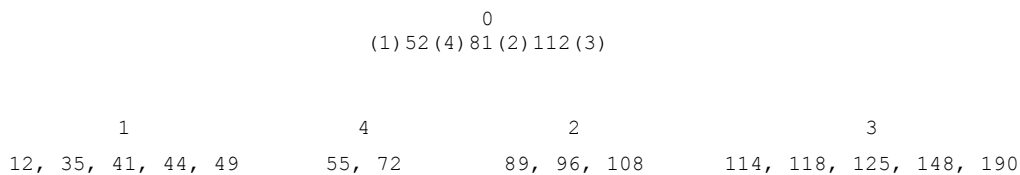
631, 914, 411, 325, 510, 549, 680, 704, 772, 236, 309, 415, 27,
208, 10

8. Repetir el ejercicio anterior pero aplicando selección natural. Comparar resultados.
9. Repetir los ejercicios de ordenación externa con selección y reemplazo y con selección natural con la siguiente secuencia de registros (representados por sus claves):
626, 909, 406, 320, 505, 544, 675, 699, 767, 231, 304, 410, 22,
203, 996

Árboles Balanceados y Archivos Indexados

Árboles B

1. Sea el Árbol B almacenado en disco que se especifica en la figura (el número superior es el número relativo de nodo, los números entre paréntesis en la raíz son números relativos de nodos sucesores; los demás son claves), de orden 4 en nodos no terminales y hasta 5 claves en las hojas (en los nodos hoja no se almacena referencia a nodos hijos nulos y se aprovecha el espacio para una clave adicional).

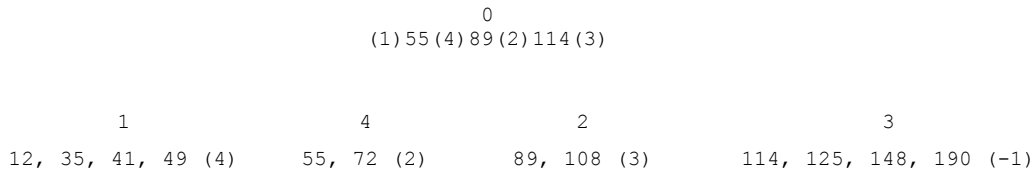


- a) Agregar la clave 130 respetando las siguientes reglas generales: la raíz debe ubicarse siempre en el nodo 0, y al partir un nodo en dos debe quedar con más carga el de la izquierda y el nodo nuevo debe quedar a la derecha.
- b) Eliminar la clave 81 y después la 118, respetando las siguientes reglas: en caso de subflujo (underflow) en un nodo, rebalancear/redistribuir con el nodo de la derecha (siempre que sea posible), y en caso de necesidad de fusión, liberar el hermano derecho.
- c) Definir tipos de datos en Pascal para implementar un árbol similar en disco pero con nodos de exactamente 256 bytes.
2. Ejemplificar un alta sin sobreflujo, un alta con sobreflujo, una baja con subflujo y sin fusión, y una baja con subflujo y con fusión en árboles B en los que se organizan números enteros de dos bytes, las referencias a nodos son enteros de dos bytes y los nodos son de exactamente dieciséis bytes. En los ejemplos los árboles iniciales deben ser de dos niveles. Determinar la cantidad de componentes en cada tipo de nodo a partir de la definición física del árbol.
3. Explicar algorítmicamente la primitiva "append" para un árbol B y enunciar casos en los que sea aplicable.
4. Dada la siguiente definición lógica de un índice
`Indice(clave, ref_reg)`
 - a) Enuncie y proporcione un ejemplo concreto de cada combinación de tipos que le quepa (identificación, clasificación, exhaustivo, selectivo, principal, alternativo), indicando para cada caso qué puede representar ref_reg en función de la organización del archivo de datos al que refiera el índice.
 - b) Proporcione una definición física para el índice estructurándolo como árbol B con nodos de 512 bytes y considerando que clave y ref_reg son valores enteros de 4 bytes cada uno.
5. Explique, proporcione definiciones físicas y analice distintas opciones para implementar índices de clasificación organizados con árboles B.

Árboles B+

6. Sea el Árbol B+ almacenado en disco que se especifica en la figura (el número superior es el número relativo de nodo, los números entre paréntesis en la raíz son números relativos de nodos sucesores, y en las hojas el número relativo del nodo siguiente; los demás son claves), de orden 4 en nodos índice y hasta

4 claves en los nodos de secuencia (en los nodos de secuencia u hojas no se almacena referencia a nodos hijos nulos y se aprovecha el espacio para una clave adicional).



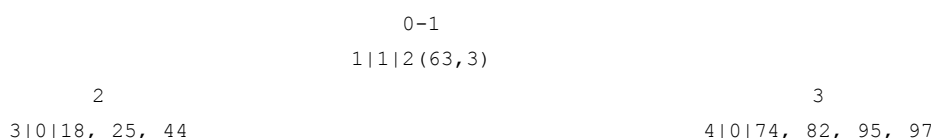
- Agregar la clave 20 respetando las siguientes reglas generales: la raíz debe ubicarse siempre en el nodo 0, y al partir un nodo en dos debe quedar con más carga el de la izquierda y el nodo nuevo debe quedar a la derecha.
 - Eliminar la clave 89 y después la 49, respetando las siguientes reglas: en caso de subflujo (underflow) en un nodo, rebalancear/redistribuir con el nodo de la derecha (siempre que sea posible), y en caso de necesidad de fusión, liberar el hermano derecho.
 - Definir tipos de datos en Pascal para implementar un árbol similar en disco pero con nodos de exactamente 128 bytes.
7. Graficar la evolución de un índice B+ con nodos de 32 bytes. Indicar para cada nodo su número relativo, y para cada entrada de índice el valor de la clave y el número relativo de registro, del cual deberá llevarse cuenta; asimismo se debe reutilizar los números de registro que se den de baja.
- Asumir que las claves son de 2 bytes, los números relativos de registro de 4 y los números relativos de nodo de 2.
- Rehacer el árbol completo toda vez que una operación afecte a más de un nodo. Realizar las siguientes operaciones (donde + implica alta, y - baja):
- +1653, +2094, +642, +840, +2203, +1145, +1315, +914, +711, +1031, -2094, +980, +500, -1145, -711
- Ejemplificar un alta sin sobreflujo, un alta con sobreflujo, una baja con subflujo y sin fusión, y una baja con subflujo y con fusión en árboles B+ en los que se organizan números enteros de dos bytes, las referencias a nodos son enteros de dos bytes y los nodos son de exactamente dieciséis bytes. En los ejemplos los árboles iniciales deben ser de dos niveles. Determinar la cantidad de componentes en cada tipo de nodo a partir de la definición física del árbol.
 - ¿Qué información mantendría en RAM para implementar una primitiva "prox" que devuelva el próximo par (clave, referencia) de un índice B+ abierto? Explicar algorítmicamente esta primitiva y enunciar casos en los que sea aplicable.
 - Defina lógica y físicamente índices B+ de identificación y de clasificación. Defina estructuras para implementarlos en Pascal de manera que los nodos sean exactamente de 512 bytes y considerando que las claves son de tipo LongInt y las referencias a nodos y a ubicaciones de registros son de tipo Integer.

Árboles B*

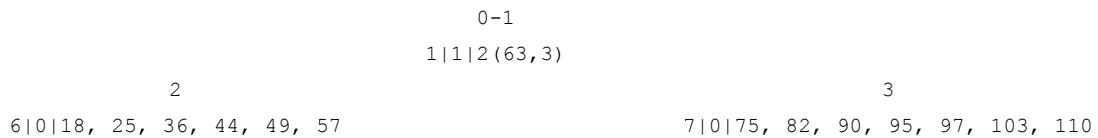
11. Dada la siguiente definición física de un árbol B* para almacenar claves enteras:
- ```

B*(cant: Byte, (nivel: Byte, hijoIzq: Byte, (clave: Word, hijoDer:
Byte)+)| ('0': Byte, (clave: Word))+)

```
- Definir tipos en Pascal para implementar el árbol con nodos de exactamente 16 bytes.
  - Asumiendo que el árbol tiene nodos de 16 bytes, eliminar las claves 18 y 82 del árbol que se bosqueja a continuación, en ese orden y redibujando el árbol para cada operación:



- c) Bajo la misma asunción que en el inciso anterior, insertar las claves 80 y 4 en el árbol que se bosqueja a continuación, en ese orden y redibujando el árbol para cada operación:



12. Ejemplificar un alta con sobreflujo y sin partición, un alta con sobreflujo y con partición, una baja con subflujo y sin fusión, y una baja con subflujo y con fusión en árboles B\* en los que se organizan números enteros de dos bytes, las referencias a nodos son enteros de dos bytes y los nodos son de exactamente dieciséis bytes. En los ejemplos los árboles iniciales deben ser de dos niveles. Determinar la cantidad de componentes en cada tipo de nodo a partir de la definición física del árbol.
13. Defina lógica y físicamente índices B\* de identificación y de clasificación. Defina estructuras para implementarlos en Pascal de manera que los nodos sean exactamente de 1024 bytes y considerando que las claves son de tipo LongInt y las referencias a nodos y a ubicaciones de registros son de tipo Integer.
14. Reevalúe el ejercicio 3 considerando los árboles B\*.
15. Defina físicamente un árbol B\* para claves alfabéticas de manera que se almacenen con longitud variable y abreviadas.

### Archivos Indexados

16. Dada la siguiente definición lógica:

```
Producto(((nombre)ie, (marca)ie)i, presentación(descripción,
existencia, precio de venta unitario)*)
```

- a) Definir físicamente al archivo considerando una organización secuencial indexada, y usando abreviaturas para el índice primario. Definir además un índice secundario por marca de producto, también usando abreviaturas.
- b) Ejemplificar el estado del archivo de datos y de los archivos de índice a partir de la creación con los siguientes registros, con bloques de 256 bytes para el archivo de datos, y nodos 64 bytes para ambos índices:

```
Leche Entera, La Serenísima, (sachet 1l, ext1, pvu1), (caja 1l, ext2,
pvu2), (caja 500cc, ext3, pvu3);
```

```
Leche Entera, Las Tres Niñas, (sachet 1l, ext1, pvu1), (caja 1l, ext2,
pvu2);
```

```
Leche parc. descr., La Serenísima, (sachet 1l, ext1, pvu1), (caja 1l,
ext2, pvu2), (caja 500cc, ext3, pvu3);
```

```
Leche parc. descr., Las Tres Niñas, (sachet 1l, ext1, pvu1), (caja 1l,
ext2, pvu2);
```

```
Leche parc. descr. c/vit. C, La Serenísima, (sachet 1l, ext1, pvu1);
```

exti y pvui, i: 1, 2,... son constantes para abstraer valores de existencias y de precios de venta que pueden representarse en el ejemplo con los mismos nombres, pero computando la cantidad de bytes que tengan según la definición física.

17. Definir lógica y físicamente (incluyendo definiciones lógicas y físicas de archivos de control) un archivo para gestionar órdenes de servicio a automóviles en el taller de una empresa concesionaria, *justificando la organización elegida y el propósito de las estructuras de control que proponga*. Tener en cuenta que el archivo debe registrar referencias al automóvil objeto de cada servicio y a su propietario, el kilometraje del auto al momento del servicio, el motivo del servicio (puede ser por revisión técnica periódica o por algún problema en particular que refiera el propietario), la fecha y turno (mañana o tarde) de entrada del vehículo, la fecha y hora pautada para la salida, la fecha y hora real de salida, una lista de intervenciones a realizar o realizadas y una lista de repuestos o insumos. Cada intervención se caracteriza por referencias a la intervención a realizar o realizada y al técnico responsable, y por el costo de mano de

obra que debe pagar el propietario del automóvil por la intervención. Cada repuesto o insumo se caracteriza por una referencia al repuesto o insumo en sí, y por el costo que debe abonar el propietario. Los costos de mano de obra de algunas intervenciones así como de algunos repuestos pueden ser 0, si el automóvil se encontrara en garantía. Las órdenes se crean por turnos solicitados telefónicamente; hay un cupo diario para la mañana y otro para la tarde para revisiones técnicas periódicas, y otros tantos para diagnóstico-presupuestación-reparación de problemas particulares: para registrar una nueva orden se debe determinar las cantidades de órdenes ya creadas de cada tipo por día y por turno, desde el turno próximo al actual. Las órdenes correspondientes a vehículos actualmente en el taller se actualizan y consultan con frecuencia, especialmente las que registren presupuestos para ser rechazados o aprobados por el propietario del automóvil. Mensualmente se debe totalizar las intervenciones por técnico (discriminando por técnico y por identificación de intervención) para liquidar comisiones, y repuestos reemplazados en garantía (con costo 0 para el propietario del automóvil) para pedir reposiciones a fábrica.

18. Definir lógica y físicamente (incluyendo definiciones lógicas y físicas de archivos de control) un archivo para gestionar préstamos de ejemplares de libros en una biblioteca, *justificando la organización elegida y el propósito de las estructuras de control que proponga*. Tener en cuenta que el archivo debe registrar referencias al ejemplar prestado y al prestatario, la fecha en la que el prestatario retiró el ejemplar, la fecha límite para devolverlo y la fecha real de devolución, ya que el archivo debe mantener el historial de préstamos. Hay distintas clases de prestatarios, y los prestatarios de la mayoría de estas clases tienen limitaciones de cantidad de ejemplares que pueden conservar en préstamo simultáneamente y de tiempo de devolución; asimismo, la cantidad de ejemplares que un prestatario puede conservar simultáneamente y el tiempo límite de devolución pueden reducirse como penalidad por devoluciones con retraso a partir de cierta cantidad de días previos a una solicitud de préstamo: para acordar un préstamo se requiere contabilizar la cantidad de ejemplares que el solicitante conserva en su poder y la cantidad de devoluciones con retraso más la cantidad de días de retraso dentro de una determinada cantidad de días previos. Diariamente se debe listar los prestatarios que están actualmente en mora para la devolución de ejemplares prestados ordenados descendientemente por el tiempo de mora.

## Archivos Directos para Datos e Índices

### Archivos Estáticos

1. Definir tipos en Pascal para implementar un archivo directo con saturación lineal con registros de longitud fija de tipo TReg tales que  $\text{SizeOf}(\text{TReg})=87$ , y bloques de exactamente 1Kb.  
Determine una cantidad adecuada de bloques a inicializar asumiendo que se almacenaría a lo sumo 800 registros en el archivo. Justifique.  
Describa precisamente los pasos a seguir para incorporar (dar de alta) un registro en el archivo.
2. Definir en pascal el tipo de datos para un archivo directo con bloques de 2Kb y registros de longitud variable con una longitud promedio estimada de 65 bytes. Los desbordes de los bloques de direccionamiento directo se ubican en bloques comunes intercalados de manera que a cada secuencia de cinco bloques consecutivos le sigue uno de desborde común. Asumiendo que los registros tienen una clave numérica, definir la función de dispersión que evite los bloques de desborde. Precisar con cuántos bloques crearía el archivo calculando que la máxima cantidad de registros que podrá tener es 370. Explique cómo buscaría un registro en el archivo, justificando los campos de control que haya definido.
3. Definir en pascal el tipo de datos para un archivo directo con bloques de 2Kb y registros de longitud fija de tipo TReg, tal que  $\text{SizeOf}(\text{TReg})=93$ . Los desbordes de los bloques de direccionamiento directo se ubican en bloques comunes al final del archivo. Precisar con cuántos bloques crearía el archivo calculando que la máxima cantidad de registros que podrá tener es 650. Explique cómo daría de baja un registro en el archivo, justificando los campos de control que haya definido.
4. Definir en pascal el tipo de datos para un archivo directo con bloques de 4Kb y registros de longitud fija de tipo TReg, tal que  $\text{SizeOf}(\text{TReg})=93$ . Los desbordes de los bloques de direccionamiento directo se ubican en bloques de desborde exclusivos de 1Kb en otro archivo. Precisar con cuántos bloques crearía el archivo calculando que la máxima cantidad de registros que podrá tener es 650. Explique cómo crearía el archivo, justificando los campos de control que haya definido.

5. Hacer una prueba para comparar las cuatro variantes de organización estática de los ejercicios previos, realizando las mismas operaciones en un archivo dimensionado para 20 registros. Planificar las operaciones de manera que el archivo llegue a contener hasta 20 registros. Considerar que los registros son de longitud fija y que entran 3 por bloque. En la variante de bloques de desborde intercalados considerar que hay uno de desborde o secundario cada grupo de tres bloques primarios, y en la variante de bloques de desborde exclusivos en archivo a parte, considerar que la capacidad de éstos es de un registro.

Repita la prueba más de una vez, inventando claves que hagan que generen problemas para una de las variantes para ver qué sucede con las otras.

### Archivos Dinámicos

6. Grafique la evolución de un archivo directo organizado en bloques con capacidad para tres registros y dispersión extensible, realizando las operaciones que se indican a continuación mediante claves de registros (clave positiva indica inserción, y negativa, supresión):  
715, 849, 123, 511, 987, 640, -849, 404, 286, 300
7. Defina físicamente (abstrayendo la estructura de los registros mediante el nombre REG) y desarrolle en forma gráfica la evolución de un archivo directo con bloques para cuatro registros de longitud fija y con dispersión dinámica, realizando altas a partir del archivo vacío y hasta que la tabla alcance tamaño 8, y luego realizando bajas hasta que la tabla alcance tamaño 4.
8. Calculando que un archivo no tendrá más de 15 registros, se definió una organización directa extensible con una tabla de dispersión inicial de 7 posiciones para apuntar a bloque con capacidad para 3 registros.
- a) Dado el siguiente estado de la organización, especificar el estado después de insertar registros con clave 67 y 13 (en el esquema los registros están representados por sus claves, y para simplificar no se denota contador de registros o espacio libre dentro de cada bloque).

Tabla:

|   |    |
|---|----|
| 0 | -1 |
| 1 | -1 |
| 2 | 1  |
| 3 | -1 |
| 4 | 0  |
| 5 | -1 |
| 6 | 2  |

Archivo:

|   |               |
|---|---------------|
| 0 | 7; 53,81,95   |
| 1 | 7; 9, 72      |
| 2 | 7; 41, 62, 76 |

- b) ¿Por qué el campo de control al comienzo de cada bloque en el estado inicial es 7? ¿Qué significa este número? Explicar algorítmicamente la baja de un registro?
- c) Detalle algorítmicamente la inserción de un registro R con clave C(R).
9. Realizar las siguientes operaciones en un índice directo con dispersión dinámica y buckets con capacidad para cuatro registros (con detalle de la información de control necesaria en un modelo a escala de implementación), y especificar las definiciones lógica y física:  
(850, 0), (255, 1), (4463, 2), (2661, 3), (41, 4), (7768, 5), (495, 6), (220, 7), (1642, 8), (551, 9), (506, 10), (214, 11)  
Elegir un bucket cualquiera y dar de baja su contenido.

### Problemas de Organización de Archivos

1. Dadas las siguientes definiciones lógicas de archivos para un negocio de alquiler de videos, y considerando la importancia de las operaciones que se enuncian, clasifíquelos como maestros o transaccionales, proponga una organización para cada archivo y defina lógicamente los archivos de control que considere necesarios, mencionando su organización física (p.e. árbol B, B+, etc., sin proveer la definición física formal):

Socios((socio)i, apellido y nombres, dirección, (tel)?, (e-mail)?, crédito, mora(fecha, monto)\*)

```
Precios((medio('cas' | 'dvd'), categoría('24' | '48' | '72'))i, precio-alq,
precio-bonif, mora-día)
Géneros(género)
Directores(director)
Intérpretes(intérprete)
Videos((código)i, título, (género)ie, ((director)*)ie, ((intérprete)*)ie,
(copias((medio, categoría)ie, estreno(año, mes), cantidad))*2)
Alquileres(((socio)ie, fecha)i, video((código)ie, medio, precio, lím-
devolución, (f-devolución)?)+)
```

Para cada alquiler se copia el precio según el medio y categoría de cada copia alquilada como valor por defecto, y se calcula la fecha límite de devolución a partir de la fecha actual, sumándole 1, 2 o 3 días según la categoría. Si el socio tiene crédito (paga por adelantado), el precio que se copia es el bonificado. También se reporta si el socio tiene deuda por mora.

La devolución de copias alquiladas en una operación puede ser en distintos días, y para cada copia devuelta se debe registrar la fecha de devolución, y en caso de que la devolución sea con mora, calcular el cargo por mora y registrárselo al socio (con signo negativo se indica que el cargo está pago, y con positivo que está adeudado, y si el socio tiene crédito se le descuenta del mismo registrando el cargo negativo).

Todos los días se hace una lista de moras superiores a 3 días, por socio, video y medio con los datos del socio, el título del video y la cantidad de días de mora, para contactarse con los socios. Los socios suelen consultar por títulos de videos por director, por algún intérprete o por combinación de medio, género y estreno.

2. Considere las siguientes definiciones lógicas de archivos para un restaurante:

```
agrupacion_productos(cod_grupo, título)
productos(cod_prod, nombre, (descripción)?, cod_grupo, (presentación,
precio)*)
comandas(mesa, mozo, cant_cubiertos, (cod_prod, presentación, cant)+)
facturas(nro_fac, mozo, fecha, cant_cubiertos, precio_cubierto, (cod_prod,
presentación, precio, cant)+)
```

- a) Defina físicamente los archivos agrupación\_productos, productos y facturas junto con las definiciones físicas de archivos de control necesarios, considerando que:
  - agrupación\_productos es secuencial por cod\_grupo con registros de long. variable
  - productos es secuencial indexado por cod\_prod con registros de long. variable y con actualizaciones
  - facturas es secuencial indexado por nro\_fac con registros de long. variable, y sin actualizaciones
- b) Proponga una organización para las comandas y defina físicamente el o los archivos propuestos. Considere que las comandas se actualizan aleatoriamente según las mesas que se están sirviendo y que se eliminan una vez cobrado el servicio a la mesa (mesa y mozo las identifican, y se mantienen temporalmente en disco por seguridad).
- c) Defina lógicamente un índice para facturas que permita calcular lo facturado por día y aclare qué tipo de índice es.
- d) Clasifique los archivos definidos (maestros, transaccionales, de trabajo o de control). Justifique por qué el archivo de facturas tiene precios, y proponga dónde almacenar el precio de referencia del cubierto.

3. Dada las definiciones lógicas de los siguientes archivos para un sistema de administración de un hotel, y considerando las operaciones que se indican, proponer una organización para cada archivo (describiendo la organización de registros) y proveer las definiciones lógicas y las justificaciones de los índices que resulten indispensables.

```
Habitación((número)hab, tipo, (idCategoría)ie, (observación)?) //tipo indica si es single,
doble con dos camas, doble matrimonial, etc.
```

Precio((idCategHab)i, precioPorDía) //La categoría de la habitación es combinación de la capacidad, determinada por el tipo, y la comodidad

Servicio((idServicio)i, descripción, precio) //Lista de precios de lavandería, tintorería, productos de frigobar, carta de bar/restaurante con entrega en habitación, y precios de regímenes diarios de pensión (desayuno, media, completa) por persona

Huésped((idH)i, (ident(tipoDoc, número))i, nombre, domicilio, (teléfono)\*, (e-mail)?)

Estadía(((idH)ie, desde)i, hasta, (alojamiento((idHab)ie, ((idH)ie)\*, (consumo((idServicio)ie, fecha, hora, precio, cantidad ))\*)+)) //La estadía se identifica por el huésped responsable y la fecha de inicio, y puede involucrar más de una habitación, con uno o más huéspedes asignados (entre los que puede estar o no el responsable) y con servicios consumidos con cargo a la habitación

Las operaciones críticas son las consultas de disponibilidad de habitaciones para reservar en función de dos fechas, el registro de reservas en Estadía indicando el responsable y las habitaciones, el check in (se registran todos los huéspedes y se asignan a las habitaciones, mediando o no reservas), el registro de consumos con cargo a una habitación, y el check out (se calcula lo que debe pagar el responsable de una estadía antes de retirarse).