

Enunciado y solución

Se tiene un fichero con información sobre contenidos digitales que ofrece una empresa de televisión a sus clientes. De los procesos a los que está sometido este fichero, se quiere optimizar los siguientes:

P	Descripción del proceso	Nº tuplas	Frecuencia
P ₁ :	UPDATE contents SET rec_date=SYSDATE WHERE rec_date = Z;	50	0.4
P ₂ :	SELECT title FROM contents WHERE id_content=X;	1	0.299
P ₃ :	SELECT DISTINCT rec_date FROM contents WHERE title= Y;	100	0.3
P ₄ :	DELETE FROM contents WHERE rec_date BETWEEN val1 AND val2;	250	0.001

Las tasas de borrado e inserción son iguales, por lo que el contenido útil del fichero se mantiene constante. El tamaño de **title** es 24 B de media (variable), mientras que **id_content** y **rec_date** son de tamaño fijo, con 9 B y 8 B respectivamente.

El fichero contiene 10^5 registros que ocupan 349B cada uno, que se encuentran almacenados en un fichero cuya organización es serial no-consecutiva (O_0) con cubos de $E_c = 2$ bloques, cabecera de 200B (información de control), y con espacio libre distribuido (10%).

Está almacenado sobre un dispositivo de acceso aleatorio cuyo tamaño de bloque es de 2KB. Se pretende comparar la eficiencia de tres organizaciones base no consecutivas (mismo diseño de cubo).

Las alternativas son las siguientes:

- O_0 : serial no consecutiva
- O_1 : secuencial no consecutiva (elegir la CO más conveniente)
- O_2 : direccionada por CD=title sobre $N=800$ con gestión de desbordamientos en área independiente de organización serial. Para esa organización se tiene $f(\text{title})=x \quad x \in [0, 799]$ que presenta una tasa de desbordamientos del 92% del total de registros.

Se pide:

- a) Calcula el coste global de la organización original y de cada una de las dos nuevas propuestas.
- b) Tomando un puntero estándar de doble precisión y de tamaño 6B (tanto para punteros internos como externos), calcúlense los parámetros de los índices arbóreos B/B+ que puedan mejorar alguna de las organizaciones anteriormente analizadas.
- c) Aplíquense los índices del apartado anterior donde proceda, obteniendo así nuevas organizaciones O_0' , O_1' y O_2' . Calcúlese el coste global de cada uno de ellas.
- d) Discútanse los resultados obtenidos, justificando que organización es más conveniente y qué organizaciones deben evitarse y por qué. Opcionalmente, el alumno puede proponer nuevas mejoras sobre la organización elegida, u otras organizaciones diferentes que el alumno estime puedan mejorar la eficiencia del sistema descrito.

Ejercicio extra: Ficheros – (a) – O₀

O₀ serial no consecutiva, habrá que calcular:

- El tamaño de cubo
- El tamaño del fichero en cubos
- Las densidades ideal, real y de ocupación
- Costes

$$T_c = \frac{(T_{bq} * E_c - T_{inf}) * (1 - ELD)}{\text{Volumen Real}} = \frac{(2048 \frac{\text{bytes}}{\text{bq}} * \frac{2\text{bq}}{\text{cubo}} - 200B) * (1 - 0,1)}{349 \text{ bytes/registro}} \rightarrow \mathbf{11 \text{ reg/cubo}}$$

$$T_{\text{fichero}} = \frac{\text{número de registros}}{T_c} = \frac{10^5 \text{ registros}}{11 \frac{\text{registros}}{\text{cubo}}} = \mathbf{9091 \text{ cubos}}$$

$$d_i = ?$$

$$d_r = \frac{\text{numReg} * \text{Inf útil}}{T_{\text{fichero}} * E_c * T_{bq}} = \frac{10^5 \text{ registros} * \frac{349 \text{ bytes}}{\text{reg}} * ?}{9091 \text{ cubos} * 2 * 2048 \text{ bytes/bq}} = ?$$

$$d_o = \frac{\text{núm. regist.}}{N * T_c} = \frac{10^5 \text{ registros}}{9091 \text{ cubos} * 11 \text{ reg/cubo}} = 0,9999 \rightarrow \mathbf{99,99\%}$$

Ejercicio extra: Ficheros – (a) – O_0

P_1 : Actualización por clave no identificativa (50 tuplas por consulta)

Lectura Full Scan + actualización de como mucho 50 cubos (caso peor)

$$C(O_0, P_1) = Sel + act = (9091 + 50) * E_c = 18282 \text{ accessos}$$

P_2 : Consulta por clave identificativa

Caso mejor / Caso peor

$$C(O_0, P_2) = \left[\frac{1 + 9091}{2} \right] * E_c = 9092 \text{ accessos}$$

P_3 : Consulta por clave no identificativa (100 registros)

Leer de principio a fin – Full Scan

$$C(O_0, P_3) = 9091 * E_c = 18182 \text{ accessos}$$

P_4 : Consulta por clave no identificativa (250 registros) y borrado

Leer de principio a fin – Full Scan y suponer cada registro en cubo distinto

$$C(O_0, P_4) = Sel + bor = (9091 + 250) * E_c = 18682 \text{ accessos}$$

$$C(O_0) = f_1 * C(O_0, P_1) + f_2 * C(O_0, P_2) + f_3 * C(O_0, P_3) + f_4 * C(O_0, P_4) = 15505 \text{ acc.}$$

Ejercicio extra: Ficheros – (a) – O_1

O_1 : Secuencial no consecutiva con elección de la CO

- Elección de la clave de ordenación
- El tamaño de cubo
- El tamaño del fichero en cubos
- ¿Área desordenada?
- Las densidades (si es posible calcularlas)
- Costes

Valoramos **rec_date** y **title**:

- **rec_date** optimiza P_1 y P_4 , pero implica cierto desorden que en el transcurso del tiempo puede degradar la organización en exceso. Además P_4 es poco significativo
- **title** optimiza P_3 solamente y el orden no degenera

$T_c \rightarrow$ Igual que en O_0

$T_{\text{fichero}} \rightarrow$ Igual que en O_0

$d_r \rightarrow ?$ (Igual que en O_0)

$d_o \rightarrow$ Igual que en O_0

P₁: Actualización por clave la clave de ordenación (rec_date) con 50 tuplas por consulta)

Se puede aplicar la búsqueda dicotómica. Hay que tener en cuenta que cuando se busca por rec_date sobre 10⁵ registros y cada valor de rec_date devuelve en término medio 50 registros, la búsqueda se hace sobre 2000 valores. Una vez encontrada hay que desplazarse para encontrar los 50 registros con el valor de rec_date buscado y posteriormente actualizar dichos registros conservando el orden, por lo que han de ser borrados y reinsertados al final en zona desordenada

$$C(O_1, P_1) = loc + bor + ins = \left[\log_2(2000 + 1) + \left(\frac{50 + 1}{11} \right) + \frac{k}{T_c} + \frac{k}{T_c} \right] * E_c$$

$$= (11 + 5 + 5 + 5) * E_c = \mathbf{52 \text{ acc.}}$$

P₂: Consulta por clave identificativa (no cambia con respecto a O₀)

Caso mejor / Caso peor

$$C(O_1, P_2) = \left[\frac{1 + 9091}{2} \right] * E_c = 9092 \text{ accessos}$$

Ejercicio extra: Fich. – (a) – O₁ CO=rec_date

P₃: Consulta por clave no identificativa title (100 registros), no cambia
Leer de principio a fin – Full scan

$$C(O_1, P_3) = 9091 * E_c = 18182 \text{ accesos}$$

P₄: Consulta por clave de ordenación (250 registros) y borrado

Se puede aplicar la búsqueda dicotómica. Considerar el número de valores distintos del atributo que son 2000, pero al ser un rango la búsqueda puede mejorar al ser 250 (5 valores de fecha) los valores recuperados sobre los 100000, sería como si tuviéramos 400. Una vez encontrado el valor hay que desplazarse para encontrar los 250 registros que por término medio nos devuelve la consulta y finalmente hay que desplazarse para encontrar los 250 registros con el valor de rec_date buscado.

$$C(O_1, P_4) = \left[\log_2(2000 + 1) + \left(\frac{250 + 1}{11} \right) + \frac{k}{T_c} \right] * E_c = (11 + 23 + 23) * E_c$$

= 114 acc. (110 acc, si consideramos 400)

$$C(O_1) = f_1 * C(O_1, P_1) + f_2 * C(O_1, P_2) + f_3 * C(O_1, P_3) + f_4 * C(O_1, P_4) = \mathbf{8194 \text{ accesos}}$$

P_1 : Actualización por clave no identificativa con 50 tuplas por consulta

Lectura Full Scan + actualización de como mucho 50 cubos (caso peor)

$$C(O_0, P_1) = Sel + act = (9091 + 50) * E_c = 18282 \text{ accessos}$$

P_2 : Consulta por clave identificativa (no cambia con respecto a O_0)

Caso mejor / Caso peor

$$C(O_1, P_2) = \left[\frac{1 + 9091}{2} \right] * E_c = 9092 \text{ accessos}$$

P_4 : Consulta por clave no identificativa (250 registros) y borrado

Leer de principio a fin – Full Scan y suponer cada registro en cubo distinto

$$C(O_0, P_4) = Sel + bor = (9091 + 250) * E_c = 18682 \text{ accessos}$$

P₃: Consulta por clave de ordenación title (100 registros)

Se puede aplicar la búsqueda dicotómica. Si recuperamos 100 valores con los 100000 registros supone 1000 valores de title que son sobre los que habría que buscar. Una vez encontrado habría que desplazarse para encontrar el resto de registros

$$C(O_1, P_3) = \left[\log_2(1000 + 1) + \left(\frac{100 + 1}{11} \right) \right] * E_c = (10 + 9) * E_c = \mathbf{38 \text{ acc.}}$$

$$C(O_1) = f_1 * C(O_1, P_1) + f_2 * C(O_1, P_2) + f_3 * C(O_1, P_3) + f_4 * C(O_1, P_4) = \mathbf{10062 \text{ acc.}}$$

Ejercicio extra: Ficheros – (a) – O₂

O₂ direccionada por CD=title sobre N=800, gestión de desbord. en área indpte de organización serial. Tasa de desbordamientos del 92% del total de registros

- El tamaño de cubo
- Tamaño del área independiente serial
- El tamaño del fichero en cubos
- Las densidades ideal, real y de ocupación
- Costes

T_c → Igual que en O₀

Además de los 800 cubos necesarios para el direccionamiento y a los que irán a parar parte de los registros, tiene que haber un área independiente para el 92% de registros que encuentran cubos llenos (92% de 10⁵, 92000 registros)

$$\mathbf{T_{\text{area desb}} = \frac{\text{nº de registros desb.}}{T_c} = \frac{92000 \text{ registros}}{11 \frac{\text{registros}}{\text{cubo}}} = \mathbf{8364 \text{ cubos (redond. arriba)}}$$

$$\mathbf{T_{\text{fichero}} = 800 + 8364 \text{ cubos} = \mathbf{9164 \text{ cubos}}}$$

Ejercicio extra: Ficheros – (a) – O₂

$$d_i = ?$$

$$d_r = \frac{\text{numReg} * \text{Inf útil}}{T_{\text{fichero}} * E_c * T_{\text{bq}}} = \frac{10^5 \text{ registros} * \frac{349 \text{ bytes}}{\text{reg}} * ?}{800 + 8364 \text{ cubos} * 2 * 2048 \text{ bytes/bq}} \rightarrow ?$$

$$d_o = \frac{\text{núm. regist.}}{N * T_c} = \frac{10^5 \text{ registros}}{(800 + 8364) * 11 \text{ reg/cubo}} = 0,9920 \rightarrow \mathbf{99,20\%}$$

P₁: Actualización por clave no identificativa (50 tuplas por consulta)

Lectura Full Scan en todo el fichero, incluyendo el área independiente + actualización de como mucho 50 cubos (caso peor)

$$C(O_2, P_1) = \text{Sel} + \text{act} = (9164 + 50) * E_c = 18428 \text{ accessos}$$

Ejercicio extra: Ficheros – (a) – O₂

P₂: Consulta por clave identificativa

Caso mejor/peor

$$C(O_2, P_2) = \left[\frac{1 + 9164}{2} \right] * E_c = 9165 \text{ accessos}$$

P₃: Consulta por clave no identificativa (100 registros)

Puede aprovechar el direccionamiento por 'title' sin embargo la alta probabilidad de que el registro haya desbordado....

$$C(O_2, P_3) = (1 + 0,92 * 8364) * E_c = 15392 \text{ accessos}$$

P₄: Consulta por clave no identificativa (250 registros) y borrado

Leer de principio a fin – Full Scan (incluida el área para desbordamientos) y suponer cada registro en cubo distinto

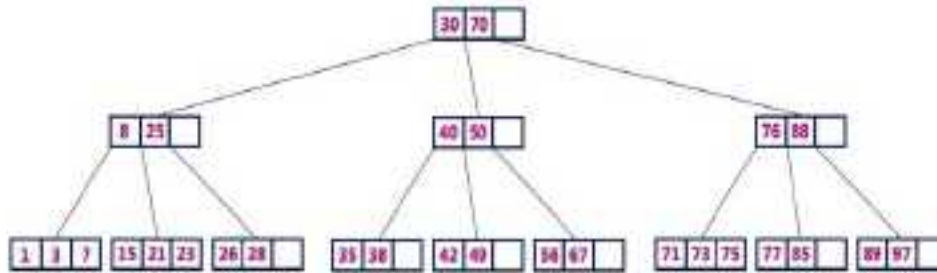
$$C(O_0, P_4) = Sel + bor = (9164 + 250) * E_c = 18828 \text{ accessos}$$

$$C(O_2) = f_1 * C(O_2, P_1) + f_2 * C(O_2, P_2) + f_3 * C(O_2, P_3) + f_4 * C(O_2, P_4) = 14748 \text{ acc.}$$

Ejercicio extra: Ficheros – (b)

Árbol B sobre id_content

¿Cómo son los nodos y el árbol?



$$m \cdot T_{\text{ptro_interno}} + k \cdot T_{\text{entrada}} < T_{\text{nodo}}$$

$$m \cdot 6 + (m - 1) \cdot (6 + 9) < 2048$$

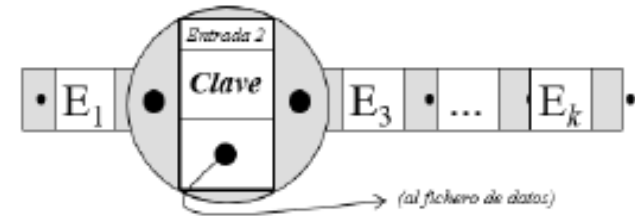
$$m \leq 2063/21$$

$$m = 98$$

$$k = 97 \text{ (puesto que } k = m - 1)$$

$$K_{\text{mín}} = k/2 = 48$$

$$m_{\text{mín}} = (m+1)/2 = 49$$



Cada nodo:

- Contiene entradas de índice (pares clave indización-puntero a los datos)
- Punteros (para apuntar nodos hijo)

Consideraciones

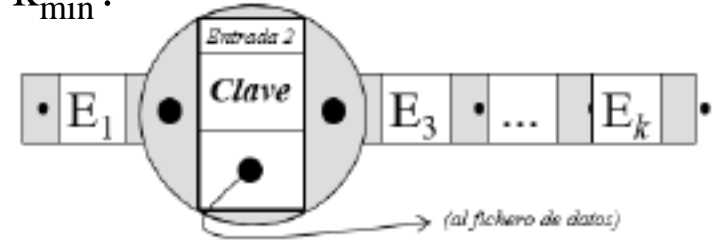
- Puntero interno de 6B
- Puntero externo de 6B
- Tamaño de la entrada: 6B del puntero externo + 9B fijos de la clave (id_content)
- $k = m - 1$
- Tomamos el tamaño de nodo más pequeño posible, es decir 1 bloque (2048 B)

Ejercicio extra: Ficheros – (b)

Árbol B sobre id_content

¿Cuántos bloques necesitamos considerando la k_{\min} ?

$$T_f(\text{índ}_K) = 10^5/48 = 2084 \text{ bloques}$$



¿Cuántos niveles necesitamos en el árbol?

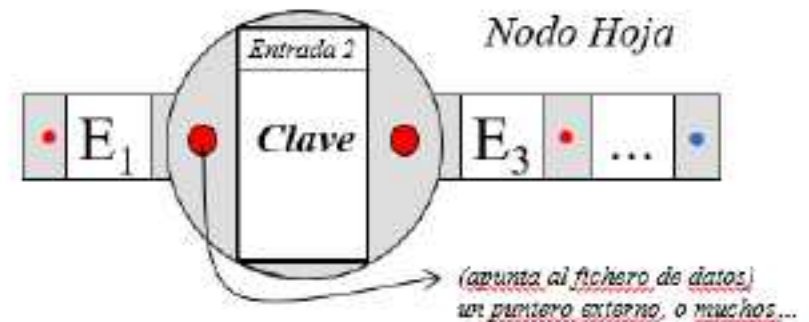
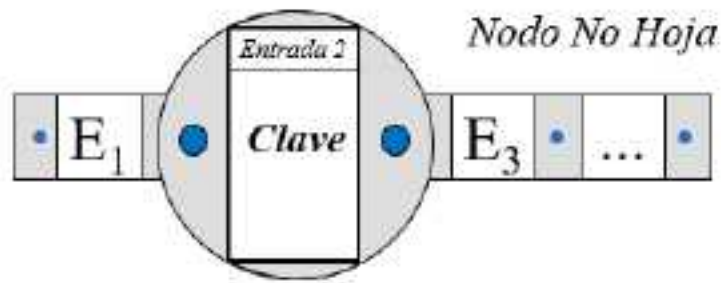
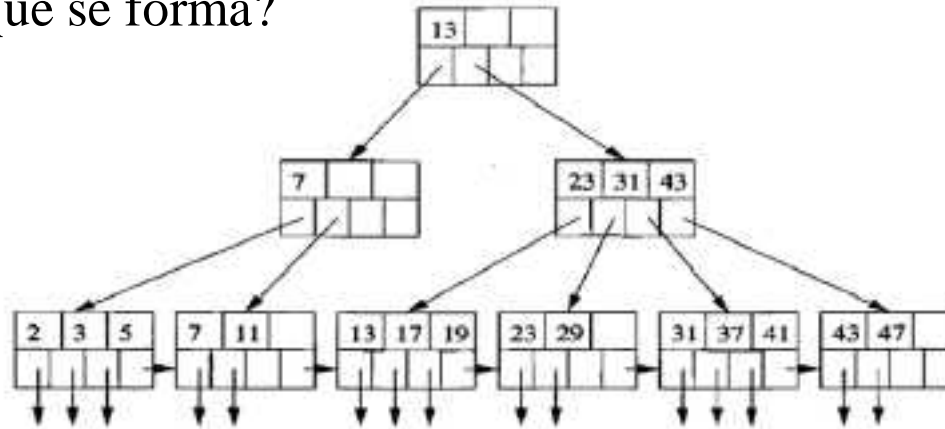
Nivel	nodos	entradas	Acumulado	
1	1	1	1	
2	2	$2 \cdot 48 = 96$	97	
3	$2 \cdot 49 = 98$	$98 \cdot 48 = 4704$	4801	
4	$98 \cdot 49 = 4802$	$4802 \cdot 48 = 230496$	235297	$> 10^5$

$n = 3$ niveles, en general 2 accesos si la raíz está en memoria

Ejercicio extra: Ficheros – (b)

Árbol B⁺ sobre title

¿Cómo es el árbol que se forma?



- En los nodos con **hijos** (**no hoja**) no hay punteros externos
- En los nodos **hoja** no hay punteros a nodo hijo, pero sí habrá punteros externos. Para apuntar a los datos, la entrada debe estar en una hoja, por lo que todas las entradas están en nodos hojas, y en los no hoja sólo hay copias discriminantes

Ejercicio extra: Ficheros – (b)

Árbol B⁺ sobre title

Orden del árbol (m): se calcula para nodos no hoja, como en árboles B, teniendo en cuenta que las entradas de esos nodos carecen de puntero externo (title tiene un tamaño de 24 bytes de media, por lo que necesita marca)

$$m \cdot T_{\text{puntero_interno}} + (m-1) \cdot T_{\text{clave}} < T_{\text{nodo}}$$

$$m \cdot 6 + (m-1) \cdot (24+1) < 2048 \rightarrow m = 66 \rightarrow m_{\min} = (m+1)/2 = 33$$

Tamaño de las entradas (marca por el n° de reg. con ese valor de title):

$$T_{\text{entrada}} = (\text{Tamaño title} + \text{Marca}) + \text{Marca} + \text{Tamaño puntero} \cdot n^{\circ} \text{ registros}$$

$$T_{\text{entrada}} = (24B + 1B) + 1B + 6B \cdot 100 = 626 B$$

Cálculo del valor de k:

$$k \cdot T_{\text{entrada}} + T_{\text{puntero_interno}} < T_{\text{nodo}}$$

$$k \cdot 626 + 6 < 2048 \rightarrow k = 3 \rightarrow k_{\min} = (k+1)/2 = 2$$

¿Cuántas hojas tendremos?

¿Hay que considerar que tenemos 1000 valores distintos de title porque se recuperan unos 100 por valor y tenemos 100000 registros

Número de hojas = $1000 / 2 = 500$ hojas (que constituyen el nivel n)

¿Cuántos nodos necesitamos en el nivel n-1?

Tendremos que considerar el m_{\min} , puesto que los nodos de este nivel serán no hoja $\rightarrow 500/33=15$

¿Cuántos nodos necesitamos en el nivel n-2?

Tendremos que considerar el m_{\min} , puesto que los nodos de este nivel serán no hoja $\rightarrow 15/33 < 1 \rightarrow$ éste será el raíz

Por tanto, el árbol tendrá 3 niveles (2 accesos a memoria con la raíz siempre)

Ejercicio extra: Ficheros – (b)

Árbol B⁺ sobre rec_date

Orden del árbol (m): se calcula para nodos no hoja, como en árboles B, teniendo en cuenta que las entradas de esos nodos carecen de puntero externo (rec_date tiene un tamaño fijo de 8 bytes)

$$m \cdot T_{\text{puntero_interno}} + (m-1) \cdot T_{\text{clave}} < T_{\text{nodo}}$$

$$m \cdot 6 + (m-1) \cdot 8 < 2048 \rightarrow m = 146 \rightarrow m_{\min} = (m+1)/2 = 73$$

Tamaño de las entradas (marca por el n° de reg. con ese valor de rec_date):

$$T_{\text{entrada}} = (\text{Tamaño rec_date}) + \text{Marca} + \text{Tamaño puntero} \cdot n^{\circ} \text{ registros}$$

$$T_{\text{entrada}} = (8B) + 1B + 6B \cdot 50 = 309 B$$

Cálculo del valor de k:

$$k \cdot T_{\text{entrada}} + T_{\text{puntero_interno}} < T_{\text{nodo}}$$

$$k \cdot 309 + 6 < 2048 \rightarrow k = 6 \rightarrow k_{\min} = (k+1)/2 = 3$$

¿Cuántas hojas tendremos?

¿Hay que considerar que tenemos 2000 valores distintos de rec_date porque se recuperan unos 50 por valor y tenemos 100000 registros

Número de hojas = $2000 / 3 = 666$ hojas (que constituyen el nivel n)

¿Cuántos nodos necesitamos en el nivel n-1?

Tendremos que considerar el m_{\min} , puesto que los nodos de este nivel serán no hoja $\rightarrow 666/73=9$

¿Cuántos nodos necesitamos en el nivel n-2?

Tendremos que considerar el m_{\min} , puesto que los nodos de este nivel serán no hoja $\rightarrow 9/73 < 1 \rightarrow$ éste será el raíz

Por tanto, el árbol tendrá 3 niveles (2 accesos a memoria con la raíz siempre)

Ejercicio extra: Fich. – (b) – Resumen

Resumen de organizaciones propuestas:

	P ₁	P ₂	P ₃	P ₄	Coste Total
O ₀	18282	9092	18182	18682	15505
O ₁ rec_date	52	9092	18182	114	8194
O ₁ title	18282	9092	38	18682	10062
O ₂	18428	9165	15392	18828	14748
f _i	0,4	0,299	0,3	0,001	

Resumen de los índices propuestos:

	B (id_content)	B+ (title)	B+ (rec_date)
Orden (m)	98	66	146
K _{min}	48	2	3
Niveles (n)	3	3	3
Número de nodos	2083	500 hojas	666 hojas

Ejercicio extra: Fich. – (c) – Aplic. Índ.

Aplicación de todos los índices propuestos a la organización serial no consecutiva

	P_1 Sel. No id. (rec_date) + act -> (50)	P_2 Sel. Ident. id_conten	P_3 Sel. No id. (title) -> (100)	P_4 Sel. No. Id. (rango rec_date) + borr -> (250)	Coste Total
O_0	18282	9092	18182	18682	15505
O_0'	Usar B+ (rec_date) $(n-1) = 2$ Leer cubos y escribirlos $50 * E_c = 100$ $50 * E_c = 100$ Reorganizar B+ 50 reg. afectados, punt en misma entrada. Cambiar la entrada de hoja. Reescribir donde estaban y leer y escribir donde han de ir $((1)+(2+1))=4$ 206	Usar B Lectura $B: (n-1) = 2$ Leer cubo $1 * E_c = 2$ 4	Usar B+ (title) $(n-1) = 2$ Leer cubos (supuestos en dist. cubos) $100 * E_c = 200$ 202	Usar B+ (rec_date) $(n-1) + 1 = 2 + 1$ (5 v. rec_date) Leer cubos (supuestos en dist. cubos) y escribirlos $250 * E_c = 500$ $250 * E_c = 500$ Reorganizar índice B+(rec_date) 250 reg. -> 5 val. Cons. 2 nodos, que hay que leer y reescribir Reorganizar índice B, cada valor puede estar en nodo dist->250, pero hay que leer suponiendo que están en distintos->2*250 1755 ¿Qué ocurre con B+(title)?	145,9
f_i	0,4	0,299	0,3	0,001	

Ejercicio extra: Fich. – (c) – Aplic. Índ.

Aplicación del índice B y del B+(title)

	P ₁ Sel. No id. (rec_date) + act ->(50)	P ₂ Sel. Ident. Por id_conten	P ₃ Sel. No id. (title) - > (100)	P ₄ Sel. No. Id. (rango rec_date) + borr -> (250)	Coste Total
O ₁ CO = rec_date	52	9092	18182	114	8194
O ₁ ' CO = rec_date	52	Usar B Lectura B: (n-1) = 2 Leer cubo 1*Ec = 2 4	Usar B+ (title) (n-1) = 2 Leer cubos (supuestos en dist. cubos) 100*Ec = 200 202	114 Reorganizar índice B, cada valor puede estar en nodo dist- >250, pero hay que leer suponiendo que están en distintos- >2*250 864 ¿Qué ocurre con B+(title)?	83,5
f _i	0,4	0,299	0,3	0,001	

Aplicación del índice B y del B+(rec_date)

	P ₁ Sel. No id. (rec_date) + act ->(50)	P ₂ Sel. Ident. Por id_conten	P ₃ Sel. No id. (title) -> (100)	P ₄ Sel. No. Id. (rango rec_date) + borr -> (250)	Coste Total
O ₁ title	18282	9092	38	18682	10062
O ₁ ' CO = title	Usar B+ (rec_date) (n-1) = 2 Leer cubos y escribirlos 50*Ec = 100 50*Ec = 100 Reorganizar B+ 50 reg. afectados, punt en misma entrada. Cambiar la entrada de hoja. Reescribir donde estaban y leer y escribir donde han de ir ((1)+(2+1))=4 206	Usar B Lectura B: (n-1) = 2 Leer cubo 1*Ec = 2 4	38	Usar B+ (rec_date) (n-1) + 1= 2 + 1 (5 v. rec_date) Leer cubos (supuestos en dist. cubos) y escribirlos 250*Ec = 500 250*Ec = 500 Reorganizar índice B+ 250 reg. -> 5 val. Cons. 2 nodos, que hay que leer y reescribir Reorganizar índice B, cada valor puede estar en nodo dist->250, pero hay que leer suponiendo que están en distintos->2*250 1755	96,8
f _i	0,4	0,299	0,3	0,001	

Ejercicio extra: Fich. – (c) – Aplic. Índ.

Aplicación de B y B+(rec_date), si además aplicmos B+(title) mejoraríamos pero no tendría sentido, al no poder superar a la O_0 '.

	P_1 Sel. No id. (rec_date) + act ->(50)	P_2 Sel. Ident. Por id_conten	P_3 Sel. No id. (title) -> (100)	P_4 Sel. No. Id. (rango rec_date) + borr -> (250)	Coste Total
O_2	18428	9165	15392	18828	14748
O_2'	Usar B+ (rec_date) $(n-1) = 2$ Leer cubos y escribirlos $50 * E_c = 200$ $50 * E_c = 200$ Reorganizar B+ 50 reg. afectados, punt en misma entrada. Cambiar la entrada de hoja. Reescribir donde estaban y leer y escribir donde han de ir $((1)+(2+1))=4$ 206	Usar B Lectura $B: (n-1) = 2$ Leer cubo $1 * E_c = 2$ 4	 Usar B+ (title) $(n-1) = 2$ Leer cubos (supuestos en dist. cubos) $100 * E_c = 200$ 202 	Usar B+ (rec_date) $(n-1) + 1 = 2 + 1$ (5 v. rec_date) Leer cubos (supuestos en dist. cubos) y escribirlos $250 * E_c = 500$ $250 * E_c = 500$ Reorganizar índice B+ 250 reg. -> 5 val. Cons. 2 nodos, que hay que leer y reescribir Reorganizar índice B, cada valor puede estar en nodo dist->250, pero hay que leer suponiendo que están en distintos->2*250 1755	 145,9
f_i	0,4	0,299	0,3	0,001	