

## 1. – DESCRIPCIÓN

Partiendo de esta especificación sobre los productos que necesitamos almacenar en un fichero.

*“La compañía dispone de un catálogo de productos, que se caracterizan por un nombre unívoco, por el tipo de café (arábica, libérica, canéphora), el varietal (Blue Mountain, Colombia, Bourbon,...), procedencia (país), tueste (natural, torrefacto, mezcla), proceso (normal o descafeinado) y por los formatos de comercialización: grano crudo, grano tostado, molido, liofilizado (soluble), en cápsulas, o preparado”*

Realiza:

## 2. – APARTADOS A REALIZAR

1. Diseño físico-lógico pésimo y calcula la densidad ideal. Teniendo en cuenta que las longitudes máximas de los campos son:

Nombre C (50), tipo C(10), varietal C(30), procedencia C(15), tueste C(10), proceso C(12) y el formato es C(20).

La información útil de cada campo es: 32'5B, 7B, 17'3B, 10B, 6'2B, 6'1B y 18'9B (donde de media cada producto posee 1'5 formatos de comercialización)

```
name b(50),  
type b(10),  
varietal b(30),  
origin b(15),  
roast b(10),  
process b(12),  
(format1 b(20))6
```

```
Tr = 50+10+30+15+10+12+6*20 = 247 B  
útil= 32.5+7+17.3+10+6.2+6.1+1.5*18.9 = 107.45 B  
di= 107.45 / 247 = 43.5%
```

2. Considerando que el  $T_{bq}=2048B$  en soporte secundario y que la organización de los registros es consecutiva. Calcula el número de accesos que tendrías que realizar para leer todos los productos (sabiendo que tenemos 750 productos según la carga que hemos realizado)

```
n = ⌈ r*Tr/Tb ⌉ = ⌈ 750*247/2048 ⌉ = 91 bloques → 91 accesos
```

3. Considera ahora que la organización es no-consecutiva con  $E_c=1$ . Realiza los mismos cálculos que en el apartado anterior.

$$T_c = \lfloor (E_c - T_{ctrl}) * (1 - ELD) / T_r \rfloor = \lfloor (2048 - 0) * (1 - 0) / 247 \rfloor = 8 \text{ reg/cubo}$$

$$N = \lceil r / T_c \rceil = \lceil 750 / 8 \rceil = 94 \text{ cubos} \rightarrow 94 \text{ accesos para una consulta a la totalidad}$$

4. Compara los accesos y el espacio utilizado para cada organización y discute las ventajas e inconvenientes.

La organización consecutiva optimiza el uso del espacio y, por lo tanto, reduce los accesos necesarios para una exploración completa. Dado que el archivo es fijo (la longitud de los registros es constante), la no consecutiva no ofrece ninguna ventaja: los huecos se reutilizan fácilmente y las actualizaciones se pueden realizar en la misma ubicación donde estaba el registro anterior (no hay movimiento de registro). El problema es que la densidad ideal es demasiado baja, lo que hace que el archivo sea muy grande (en este ejercicio el volumen es bajo, sólo 750 registros, por lo que el archivo es muy pequeño y hasta cabe en memoria; pero con volúmenes más grandes, pronto será ineficiente).

5. Realiza los mismos apartados (1-4) para el diseño físico-lógico óptimo.

Primera versión:

```

name_len b(1),
name b(name_len),
type b(1),
varietal_len b(1),
varietal b(varietal_len),
origin_len b(1),
origin b(origin_len),
roast b(1),
process b(1),
num_formats b(1),
(format b(1)) num_formats
  
```

$$T_r = 1 + 32.5 + 1 + 1 + 17.3 + 1 + 10 + 1 + 1 + 1 + 1.5 * 1 = 68.3 \text{ B}$$

$$\text{útil} = 32.5 + 1 + 17.3 + 10 + 1 + 1 + 1.5 * 1 = 64.3 \text{ B}$$

$$d_i = 64.3 / 68.3 = 94.14\%$$

Comentarios: hemos introducido tres marcas de longitud y una marca de reiteración; además, hemos codificado como enumerados los campos tipo, tueste, proceso y formato.

Sin embargo, el formato puede ser codificado por un mapa (seis valores → seis bits → un byte) que admite la multivaluación, evitando la necesidad de una marca de reiteración. Además, podemos codificar conjuntamente el tipo (3 valores), el tostado (3 valores) y el proceso (2 valores) en un solo mapa (8 bits → 1 byte).

#### Segunda versión:

```
name_len b(1),  
name b(name_len),  
varietal_len b(1),  
varietal b(varietal_len),  
origin_len b(1),  
origin b(origin_len),  
map_type b(1),  
map_format b(1),
```

$$T_r = 1 + 32.5 + 1 + 17.3 + 1 + 10 + 1 + 1 = 64.8 \text{ B}$$

$$\text{útil} = 32.5 + 17.3 + 10 + 1 + 1 = 61.8 \text{ B}$$

$$d_i = 61.8 / 64.8 = 95.37\%$$

#### Fichero serial consecutivo:

$$n = \lceil r \cdot T_r / T_b \rceil = \lceil 750 \cdot 64.8 / 2048 \rceil = 24 \text{ bloques} \rightarrow 24 \text{ accesos}$$

#### Fichero serial No consecutivo:

$$T_c = \lfloor (T_b - T_{ctrl}) \cdot (1 - ELD) / T_r \rfloor = \lfloor (2048 - 0) \cdot (1 - 0) / 64.8 \rfloor = 31 \text{ reg/cubo}$$

$$N = \lceil r / T_c \rceil = \lceil 750 / 31 \rceil = 24 \text{ cubos} \rightarrow 24 \text{ accesos para recorrido a la totalidad}$$

Comentarios: ahora tenemos registros de tamaño variable, y existen ciertas ventajas para la no consecutividad (como la actualización en la misma ubicación). Como los registros son bastante pequeños, el uso del espacio es bastante similar (eventualmente, los archivos son del mismo tamaño). Sin embargo, se pueden hacer mejoras, como introducir espacio libre distribuido para la actualización (pct\_free), garantizar una ocupación mínima, etc.