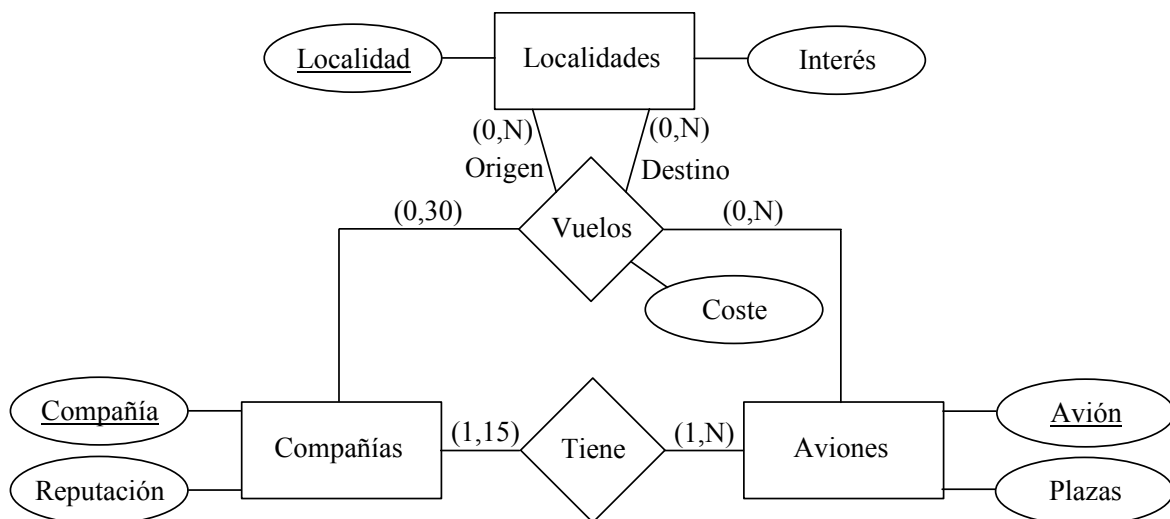


**Examen de Ficheros y bases de datos (cód. 520)**  
**Ingeniería Técnica en Informática de Gestión**  
**Convocatoria de junio**  
**I PARCIAL**

- 1) (4,35 puntos) Una agencia de viaje oferta vuelos a sus clientes. Cada vuelo tiene un origen, un destino, una compañía aérea y diferentes modelos de aviones. Los orígenes y destinos son localidades de las que se desea saber el nombre de la localidad y una indicación para describir su atractivo turístico. De las compañías aéreas se debe conocer su nombre y su reputación (por ejemplo, su capacidad para perder maletas). Finalmente, de los aviones se desea saber el modelo y su capacidad. Cada vuelo tiene un coste que depende de estos cuatro factores (origen, destino, compañía y modelo de avión). Cada compañía ofrece a lo sumo 30 vuelos. Una compañía tiene entre 1 y 15 modelos de aviones. Cada modelo de avión debe ser ofertado por al menos una compañía. Si un origen y un destino están cubiertos por una compañía, debe ofrecer al menos dos modelos de aviones. Si una compañía ofrece un vuelo de ida, debe ofrecer un vuelo de vuelta, aunque no necesariamente en el mismo modelo de avión. Se pide:
- a) (1,85 puntos) Diseñar el diagrama entidad-relación con el mínimo número de atributos necesarios (no introducir identificadores artificiales) y sin usar generalizaciones. Subrayar las clave primarias y marcar las restricciones posibles. Indicar cuáles no se pueden representar en el modelo E/R y por qué.

**Solución:**



**Restricciones:**

- Cada compañía ofrece a lo sumo 30 vuelos. Participación máxima de 30 de Compañías en Vuelos. Se indica una mínima de uno porque no tendría sentido tener una compañía aérea que no ofrezca vuelos.
- Una compañía tiene entre 1 y 15 modelos de aviones. Participación (1,15) de Compañías en Tiene.
- Cada modelo de avión debe ser ofertado por al menos una compañía. Participación (1,N) de Aviones en Tiene.
- Si un origen y un destino están cubiertos por una compañía, debe ofrecer al menos dos modelos de aviones. No es posible expresarlo con clave, ni con participación ni con cardinalidad, que son los únicos tipos de restricciones disponibles en el modelo E/R.
- Si una compañía ofrece un vuelo de ida, debe ofrecer un vuelo de vuelta, aunque no necesariamente en el mismo modelo de avión. Igual que el anterior.
- No hay restricciones sobre las localidades, así que la participación de los orígenes y los destinos en Vuelos es (1,N).

- b) (1,25 puntos) Transformar el diagrama entidad-relación en el modelo relacional (sin simplificaciones). Expresar algebraicamente las restricciones de integridad referencial.

**Solución:**

*Tipos de entidades:*

- Localidades(Localidad, Interés)
- Compañías(Compañía, Reputación)
- Aviones(Avión, Plazas)

*Tipos de relaciones:*

- Tiene(Compañía, Avión)
- Vuelos(Origen, Destino, Compañía, Avión)

*Restricciones de integridad referencial:*

- $\pi_{Compañía}(Tiene) \subseteq \pi_{Compañía}(Compañías)$
- $\pi_{Avión}(Tiene) \subseteq \pi_{Avión}(Aviones)$
- $\pi_{Origen}(Vuelos) \subseteq \pi_{Localidad}(Localidades)$
- $\pi_{Destino}(Vuelos) \subseteq \pi_{Localidad}(Localidades)$
- $\pi_{Compañía}(Vuelos) \subseteq \pi_{Compañía}(Compañías)$
- $\pi_{Avión}(Vuelos) \subseteq \pi_{Avión}(Aviones)$

- c) (1,25 puntos) Indicar el número de accesos necesarios para resolver la consulta "Determinar los modelos de aviones y su número de plazas que cubren un trayecto de un origen a un destino determinados" en término de los volúmenes de las entidades y relaciones, asumiendo que todas las localidades están conectadas con vuelos en ambos sentidos y con una distribución uniforme de aviones (cada origen y destino está cubierto en media por el mismo número de aviones).

**Solución:**

Tabla de accesos:

Concepto	Tipo	Accesos	Tipo	Motivo
Vuelos	R	$n$	L	Para averiguar los vuelos entre un origen y un destino y el modelo de avión que cubre cada uno
Aviones	R	$n$	L	Para determinar el número de plazas de cada avión

Leyenda: E=Entidad, R=Relación, L=Lectura

$n$  es el número de vuelos entre un origen y un destino determinados. Dado un volumen  $V_L$  para Localidades y un volumen  $V_V$  para Vuelos, y asumiendo que todas las localidades están conectadas con vuelos en ambos sentidos y con una distribución uniforme para Vuelos, el número de aviones que conecta un origen con un destino se calcula como  $|V_V|$  dividido por el número de pares  $\langle \text{origen, destino} \rangle$ . Este número de pares son las variaciones sin repetición de  $|V_L|$  elementos tomados de dos en dos, es decir:

$$V_{|V_L|}^2 = \frac{|V_L|!}{(|V_L| - 2)!}$$

Por tanto, el número  $n$  de accesos se calcula como:

$$n = \frac{|V_V| (|V_L| - 2)!}{|V_L|!}$$

- 2) (5,65 puntos) Dadas las siguientes tablas relacionales:

$$R(A,B,C) = \{ \langle 0,0,0 \rangle, \langle 1,1,2 \rangle \}$$

$$S(B,D,E) = \{ \langle 0,0,0 \rangle, \langle 1,1,0 \rangle \}$$

$$T(C,D,F) = \{ \langle 0,0,0 \rangle \}$$

Se pide:

- a) (1,25 puntos) Expresar la equireunión (natural) de R y S con operaciones básicas del álgebra relacional (sin considerar la join), con SQL, con cálculo relacional de tuplas y con cálculo relacional de dominios. Indicar las tuplas resultado.

**Solución:**

Si queremos crear las tablas para probar las consultas:



**CREATE TABLE R (A INTEGER PRIMARY KEY, B INTEGER, C INTEGER);**  
**CREATE TABLE S (B INTEGER PRIMARY KEY, D INTEGER, E INTEGER);**  
**CREATE TABLE T (C INTEGER PRIMARY KEY, D INTEGER, F INTEGER);**

Introducimos algunos datos:

**INSERT INTO R VALUES (0,0,0); A,B,C**  
**INSERT INTO R VALUES (1,1,2);**  
**INSERT INTO S VALUES (0,0,0); B,D,E**  
**INSERT INTO S VALUES (1,1,0);**  
**INSERT INTO T VALUES (0,0,0); C,D,F**  
**INSERT INTO T VALUES (0,2,0);**

Las consultas que nos piden:

AR:  $\pi_{R.A,R.B,R.C,S.D,S.E}(\sigma_{R.B=S.B}(R \times S))$

SQL: **SELECT \* FROM R NATURAL INNER JOIN S;**

CRT:  $\{t \mid \exists r (r \in R \wedge r[A]=t[A] \wedge r[B]=t[B] \wedge r[C]=t[C] \wedge \exists s (s \in R \wedge s[B]=t[B] \wedge s[D]=t[D] \wedge s[E]=t[E]))\}$

CRD:  $\{ \langle a,b,c,d,e \rangle \mid (\langle a,b,c \rangle \in R \wedge \langle b,d,e \rangle \in S) \}$

El resultado:

B	A	C	D	E
0	0	0	0	0
1	1	2	1	0

b) (0,60 puntos) Expresar  $(R \bowtie S) \bowtie T$  en SQL estándar. Indicar las tuplas resultado.

**Solución:**

**SELECT \* FROM**  
**(R NATURAL LEFT OUTER JOIN S)**  
**NATURAL RIGHT OUTER JOIN T;**

El resultado:

C	D	B	A	E	F
0	0	0	0	0	0

c) (0,85 puntos) Encontrar mediante SQL todos los valores de C distintos en R tales que no existe una tupla en S con el mismo valor para S.E. Indicar las tuplas resultado.

**Solución:**

**SELECT DISTINCT C**  
**FROM R**  
**WHERE NOT EXISTS**  
**(SELECT \* FROM S WHERE R.C=S.E);**

El resultado:

C
2



- d) (0,85 puntos) Encontrar mediante SQL todos los valores de B distintos en R tales que al menos existe una tupla en S con el mismo valor para S.D pero distinto del valor de cualquier D de T. Indicar las tuplas resultado.

**Solución:**

```
SELECT DISTINCT B
FROM R
WHERE
  EXISTS (SELECT * FROM S WHERE R.B=S.D) AND
  NOT EXISTS (SELECT * FROM T WHERE R.B=T.D);
```

El resultado:

```
      B
-----
      1
```

- e) (0,6 puntos) Indicar todas las dependencias funcionales que se podrían imponer sobre R sin contar las derivadas de la clave ni de las triviales.

**Solución:**

De un atributo:

$B \rightarrow A, B \rightarrow C, C \rightarrow A, C \rightarrow B$

De dos atributos:

$BC \rightarrow A$

- f) (0,6 puntos) Dada la relación  $V=R \bowtie S \bowtie T$  y el conjunto de dependencias funcionales  $F=\{A \rightarrow BC, E \rightarrow D, DB \rightarrow A\}$ , determinar las claves de V.

**Solución:**

El esquema de V es (A,B,C,D,E,F). Las claves deben contener al menos E y F dado que estos atributos no aparecen en el consecuente de ninguna dependencia funcional:

$\{A,E,F\}^+ = \{A,B,C,D,E\}$  Clave. Cualquier superconjunto de  $\{A,E,F\}$  también es clave

$\{B,E,F\}^+ = \{B,E,D,A,C\}$  Clave. Cualquier superconjunto de  $\{B,E,F\}$  también es clave

$\{C,E,F\}^+ = \{C,E,D\}$

$\{D,E,F\}^+ = \{D,E\}$

- g) (0,9 puntos) Dada la relación V y el conjunto de dependencias funcionales F anteriores, encontrar un recubrimiento mínimo.

**Solución:**

Paso 1:

Sólo aplicable a  $A \rightarrow BC$

$T=\{A \rightarrow B, A \rightarrow C, E \rightarrow D, DB \rightarrow A\}$

Paso 2:

Sólo aplicable a  $DB \rightarrow A$

¿Se puede eliminar D? Es decir, ¿ $B \rightarrow A \in T^+$ ? Para verlo:

$\{B\}^+ = \{B\}$  No.

¿Se puede eliminar B? Es decir, ¿ $D \rightarrow A \in T^+$ ? Para verlo:

$\{D\}^+ = \{D\}$  No.

Paso 3:

Ninguna de las dependencias funcionales se puede eliminar porque en el consecuente de cada una de ellas aparece un atributo que no aparece en ningún otro consecuente.

Resultado:  $T=\{A \rightarrow B, A \rightarrow C, E \rightarrow D, DB \rightarrow A\}$



**Examen de Ficheros y bases de datos (cód. 520)**  
**Ingeniería Técnica en Informática de Gestión**  
**Convocatoria de junio**  
**II PARCIAL**

- 1) (2,0 puntos) Descomponer la relación  $R(A,B,C,D)$  en relaciones que cumplan 3FN dado el conjunto de dependencias funcionales  $\{AC \rightarrow B, B \rightarrow D\}$

**Solución:**

Se comprueba si ya está en 3FN:

Para toda  $X \rightarrow Y$ , o bien  $X$  es superclave o  $Y$  contiene algún atributo que pertenece a una clave candidata.

No hay claves candidatas de un atributo

$$\{AB\}^+ = \{A, B, D\}$$

$$\{AC\}^+ = \{A, B, C, D\} \Rightarrow \text{Clave candidata}$$

$$\{AD\}^+ = \{A, D\}$$

$$\{BC\}^+ = \{B, C, D\}$$

$$\{BD\}^+ = \{B, D\}$$

$$\{CD\}^+ = \{C, D\}$$

Para  $AC \rightarrow B$ ,  $AC$  es superclave.

Para  $B \rightarrow D$ ,  $B$  no es superclave y  $D$  no pertenece a ninguna clave candidata.

Por tanto,  $R$  no está en 3FN.

Se aplica el algoritmo de descomposición en 3FN:

1. Encontrar un recubrimiento mínimo  $T$  para  $S$ .
2. Para cada  $X \rightarrow Y \in T$  crear un esquema en  $D$  con atributos  $\{X \cup \{A_1\} \cup \dots \cup \{A_k\}\}$ , donde  $X \rightarrow \{A_1\}, \dots, X \rightarrow \{A_k\}$  son todas las dependencias funcionales en  $T$  cuya parte izquierda es  $X$ . (Por tanto,  $X$  es clave para el nuevo esquema).
3. Poner el resto de atributos en una única relación para asegurar la preservación de atributos.

1.  $S$  es minimal porque  $B \rightarrow D$  sólo contiene un atributo en el antecedente y consecuente, y porque de  $AC \rightarrow B$  no se puede eliminar  $A$  o  $B$  sin eliminar su cierre, dado que ni  $A$  ni  $B$  son superclaves y  $AB$  sí.

2.  $AC \rightarrow B: \{A, B, C\}$

$B \rightarrow D: \{B, D\}$

Dado que el primer registro contiene una clave candidata del esquema  $R$ , no es necesario agregar nada más. La descomposición resultante es:

$R1 \{A, B, C\}$

$R2 \{B, D\}$

Las tablas se encuentran en 3FN.

- 2) (3,0 puntos) Se desea indexar una tabla  $r(\underline{A}: \text{String}(20), B: \text{String}(250), C: \text{String}(350))$  de 100.000 filas bajo su clave primaria mediante una estructura de índice primario denso y asignación enlazada. Los bloques tienen un tamaño de 50 bytes, sus direcciones ocupan 4 bytes y tienen mapas de bits de existencia para los registros. Se pide:

a) (0,5 puntos) Describir la estructura y tamaño de los ficheros de índice y de datos.

**Solución:**

Fichero de índices:

Registro:

Clave: 20 bytes.

Dirección: 4 bytes.

Tamaño:  $(20+4)*100.000=2.400.000$  bytes

Estructura del bloque:

Mapa de bits de existencia.

N registros.  
Dirección del siguiente bloque.

Fichero de datos:

Estructura del registro:

A: 20 bytes.

B: 250 bytes.

C: 350 bytes.

Tamaño:  $(20+250+350)*100.000=62.000.000$  bytes

Estructura del bloque:

Mapa de bits de existencia.

M registros.

Dirección del siguiente bloque.

b) (0,5 puntos) Calcular el factor de bloqueo de los ficheros de datos y de índice.

### Solución:

Para el fichero de datos:

$N*(20+250+350)*8+4*8+N \leq 50*8$ , donde:

- $N*(20+250+350)*8=N*4.960$  es el tamaño en bits ocupado por un registro (20 bytes para el campo A, 250 para el campo B y 350 para el campo C).
- $4*8=32$  es el número de bits ocupado por la dirección del siguiente bloque para la asignación enlazada
- N son los bits necesarios para el mapa de bits de existencia
- $50*8=400$  es el número de bits en los 50 bytes de un bloque

$N*(4960+1) \leq 400-32$

$N \leq 0,0742$

Por tanto, el factor de bloqueo N del fichero de datos es de 0,0742 registros por bloque. Calculando su inversa, se obtiene que son necesarios 13,5 bloques para alojar un registro. Para bloques fijos (la situación más habitual), se usan 14 bloques y se desperdicia medio por cada registro.

Para el fichero de índice:

$N*(20+4)*8+4*8+N \leq 50*8$ , donde:

- $N*(20+4)*8=N*192$  es el tamaño en bits ocupado por un registro (20 bytes para el campo clave A y 4 bytes para la dirección de bloque).
- $4*8=32$  es el número de bits ocupado por la dirección del siguiente bloque para la asignación enlazada
- N son los bits necesarios para el mapa de bits de existencia
- $50*8=400$  es el número de bits en los 50 bytes de un bloque

$N*(192+1) \leq 400-32$

$N \leq 1,9$

Por tanto, el factor de bloqueo N del fichero de índice es de 1 registro por bloque.

c) (2 puntos) Calcular el tiempo medio de acceso a una tupla en concreto bajo su valor de la clave en dos supuestos: sin usar el fichero de índices y usándolo. Determinar la ganancia de velocidad en este último caso. Considérese un tiempo de lectura de bloque de 12 ms.

### Solución:

Sin usar el fichero de índices:

En media hay que recorrer  $100.000/2=50.000$  registros del fichero de datos secuencialmente, es decir,  $50.000*14 = 700.000$  bloques. A 12ms por bloque se invierten:  $700.000*12ms=8.400 s = 140 m = 2$  horas y 20 minutos

Usando el fichero de índices:

Se recorren 50.000 registros del fichero de índices secuencialmente, es decir, 50.000 bloques. Despreciando el acceso al bloque de datos, a 12ms por bloque se invierten:  $50.000*12ms=600 s = 10 m$ .

3) (2,5 puntos) Implementar en SQL las tablas  $r(\underline{A}: \text{INTEGER}, B: \text{INTEGER})$  y  $s(\underline{C}: \text{INTEGER}, D: \text{INTEGER})$  definiendo sólo restricción de clave primaria. Para estas tablas se debe cumplir  $\pi_B(r) \subseteq \pi_C(s)$ , y se pide



programar un disparador que compruebe esta restricción de integridad referencial cuando se inserte o modifique una tupla en r y cuando se borre o modifique en s. Si al insertar o modificar en r no se encuentra correspondencia en s, se debe generar una excepción y mostrar un mensaje de error (pero permitiendo la acción). Se debe hacer igual si al borrar en s quedan tuplas colgantes en r. Cuando se modifique en s hay que actualizar en cascada las tuplas de r.

### Solución:

Las tablas vendrían definidas por:

```
CREATE TABLE r(A INTEGER PRIMARY KEY, B INTEGER);
CREATE TABLE s(C INTEGER PRIMARY KEY, D INTEGER);
```

El disparador que maneja las inserciones y modificaciones sobre r:

```
CREATE OR REPLACE TRIGGER ir_r
  BEFORE INSERT OR UPDATE ON r
  FOR EACH ROW
DECLARE
  v_B INTEGER;
  sin_correspondencia EXCEPTION;
BEGIN
  IF INSERTING OR UPDATING THEN
    SELECT COUNT(C) INTO v_B FROM s WHERE s.C=:NEW.B;
    IF v_B = 0 THEN
      RAISE sin_correspondencia;
    END IF;
  END IF;
EXCEPTION
  WHEN sin_correspondencia THEN
    DBMS_OUTPUT.PUT_LINE('No hay correspondencia en la columna C de la tabla
s para el valor '||:NEW.B||' insertado o modificado en la columna B de r.');
```

El disparador que maneja los borrados sobre s:

```
CREATE OR REPLACE TRIGGER ir_s_d
  BEFORE DELETE ON s
DECLARE
  CURSOR c_r IS
    SELECT A,B FROM r WHERE NOT EXISTS (SELECT s.C FROM s WHERE s.C=r.B);
  v_A r.A%TYPE;
  v_B r.B%TYPE;
  sin_correspondencia EXCEPTION;
BEGIN
  OPEN c_r;
  LOOP
    FETCH c_r INTO v_A,v_B;
    EXIT WHEN c_r%NOTFOUND;
    BEGIN
      RAISE sin_correspondencia;
    EXCEPTION
      WHEN sin_correspondencia THEN
        DBMS_OUTPUT.PUT_LINE('La tupla de r <'||v_A||','||v_B||'> queda
colgante porque ya no hay correspondencia con la columna C de la tabla s al
borrarse la tupla correspondiente de s.');
```

```
END;  
/
```

El disparador que maneja las modificaciones sobre *s* como modificaciones en cascada sobre *r*:

```
CREATE OR REPLACE TRIGGER ir_s_u  
  BEFORE UPDATE ON s  
  FOR EACH ROW  
DECLARE  
END;  
/
```

Nota: esta solución es correcta si los tres disparadores no están activos simultáneamente. Si lo están, se crea un ciclo de disparos que no se puede resolver porque las tablas están mutando (análogo al caso del problema 2 del segundo parcial del examen de septiembre de 2005 de BDSI <http://www.fdi.ucm.es/profesor/fernan/DBD/examenseptiembre05.zip>).

- 4) (1,0 puntos) Dadas las tablas *r*(a, b) y *s*(c, d), estímesese el tamaño y coste de la evaluación de la siguiente consulta, asumiendo una distribución aleatoria para los campos *r.a* y *s.d*, que tienen el mismo tipo de datos.

```
SELECT *  
FROM r, s  
WHERE r.a ≥ s.d;
```

#### Solución:

Esta consulta SQL se puede escribir en álgebra relacional como:

$$\sigma_{r.a \geq s.d}(r \times s)$$

Para una distribución aleatoria, la mitad de tuplas de *r* cumplirán que su campo *a* es mayor o igual que el campo *d* de la mitad de las tuplas de *s*, por lo tanto, el tamaño del resultado se puede estimar:

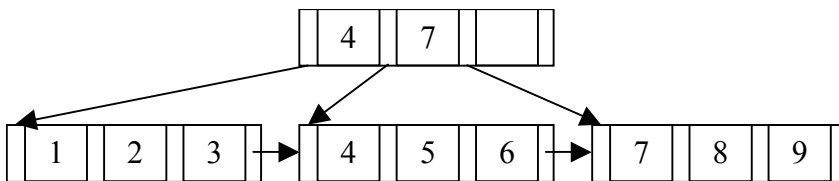
$$\frac{n_r}{2} \cdot \frac{n_s}{2} = \frac{n_r \cdot n_s}{4}$$

Si el resultado del producto cartesiano cabe en memoria, entonces el coste (número de lecturas de bloque) se calcula como:

$$\left\lceil \frac{n_r \cdot n_s}{f_r \cdot f_s} \right\rceil$$

- 5) (1,5 puntos) A partir de los nueve valores clave 1, 2, 3, ..., 9 y con  $n=4$  (número máximo de hijos por nodo) se pide:
- a) (0,5 puntos) Crear un árbol B+ con estos valores, indicando las restricciones que se deben cumplir.

#### Solución:



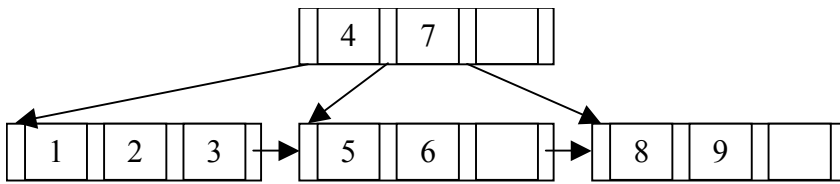
Restricciones:

- El nodo raíz tiene entre 1 y 4 hijos.
- Cada nodo interno (no hoja) tiene entre  $\lceil 4/2 \rceil = 2$  y 4 hijos.
- Los nodos hoja contienen entre  $\lceil (4-1)/2 \rceil = \lceil 1,5 \rceil = 2$  y  $4-1 = 3$  valores.

- b) (0,5 puntos) Crear un árbol B con estos valores.



**Solución:**



c) (0,5 puntos) ¿Es mínimo el número de nodos en los árboles obtenidos en los apartados anteriores?

**Solución:**

En el caso (a) sí porque se necesitan al menos 3 nodos hoja para albergar 9 valores ( $3 \cdot 3$ ) y basta con un nodo raíz para apuntar a ellos.

En el caso (b) también porque aunque los valores podrían caber en sólo tres nodos, no se cumpliría la restricción sobre el número de hijos (el nodo raíz tendría que tener cuatro hijos y sólo tendría dos).