

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

Aula y fecha: \_\_\_\_\_ **LUNES 9:00 A****Tiempo: 60 min Puntuación: 6 puntos****1) Relaciones (0/1/2 puntos)**

- *milita* : los partidos pueden tener militantes
- *entrena* : los equipos pueden tener un único entrenador
- *pilota* : los pilotos pertenecen siempre a una y sólo una escudería
- *pertenece* : el dueño de una escudería siempre es uno y sólo uno, y en todo momento tienen una y sólo una escudería.
- *local* : los equipos siempre tienen un único estadio a su disposición, y los estadios pueden tener un único equipo local.

Rellena, de acuerdo a las sentencias anteriores, con sus valores correspondientes las siguientes expresiones:

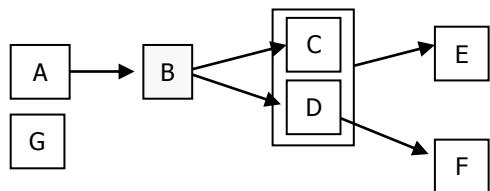
$$\text{Card}(\text{partido}, \text{milita}) = ( \textcolor{red}{0} , \textcolor{red}{N} ) \quad \text{Card}(\text{militante}, \text{milita}) = ( \textcolor{red}{0} , \textcolor{red}{N} )$$

$$\text{Card}(\text{equipo}, \text{entrena}) = ( \textcolor{red}{0} , \textcolor{red}{1} ) \quad \text{Card}(\text{entrenador}, \text{entrena}) = ( \textcolor{red}{0} , \textcolor{red}{N} )$$

$$\text{Card}(\text{piloto}, \text{pilota}) = ( \textcolor{red}{1} , \textcolor{red}{1} ) \quad \text{Card}(\text{escudería}, \text{pilota}) = ( \textcolor{red}{0} , \textcolor{red}{N} )$$

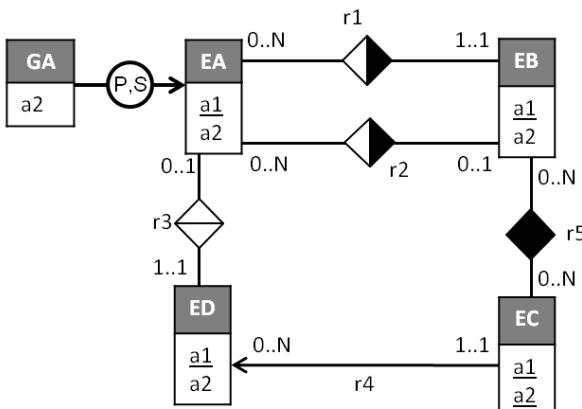
$$\text{Card}(\text{dueño}, \text{pertenece}) = ( \textcolor{red}{1} , \textcolor{red}{1} ) \quad \text{Card}(\text{escudería}, \text{pertenece}) = ( \textcolor{red}{1} , \textcolor{red}{1} )$$

$$\text{Card}(\text{equipo}, \text{local}) = ( \textcolor{red}{1} , \textcolor{red}{1} ) \quad \text{Card}(\text{estadio}, \text{local}) = ( \textcolor{red}{0} , \textcolor{red}{1} )$$

**2) Normaliza hasta 3FN o FNBC (recuerda: tablas completamente definidas) (0/1 puntos)**

R(A,G)	CP: (A,G)	CAj: A → R1
R1(A,B)	CP: A	CAj: B → R2
R2(B,C,D)	CP: B	CAj: (C,D) → R3
R3(C,D,E)	CP: (C,D)	CAj: D → R4
R4(D,F)	CP: D	

### 3) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)



GA(a1,a2)	CP: a1	CAj: a1 → EA
EA(a1,a2)	CP: a1	
ED(a1,a2,a3)	CP: a1	CAlt: a3      CAj: a3 → EA
EB(a1,a2,a3,a4)	CP: a1	VNN: a3      CAj: a3 → EA      CAj: a4 → EA
EC(a1,a2,a3)	CP: (a1,a2,a3)	CAj: a3 → ED      CAj: a1 → EB
R5(a1,a2,a3,a4)	CP: (a1,a2,a3,a4)	CAj: (a2,a3,a4) → EC      CAj: a1 → EB

### 4) Dado el siguiente esquema y estado de base de datos (0/1 punto)

**PRIMERA(p1,p2) CP(p1)**

**SEGUNDA(s1,s2,aP) CP(s1) CAj(aP)>>PRIMERA en borrados ANULAR modificaciones PROPAGAR**

**TERCERA(t1,aS,t2) CP(t1,aS) CAj(aS)>>SEGUNDA en borrados PROPAGAR modificaciones RECHAZAR**

PRIMERA		SEGUNDA			TERCERA		
p1	p2	s1	s2	aP	t1	aS	t2
1	aa	100	bb	1	500	100	aa
2	aa	200	cc	1	600	100	cc
3	bb	300	bb	2	600	200	cc

**Después de ejecutar todas las órdenes siguientes:**

delete from primera where p1=1;

update primera set p1=20 where p1=2;

delete from segunda where s1=200;

**¿Cuál sería el estado final de la BD?**

PRIMERA		SEGUNDA			TERCERA		
p1	p2	s1	s2	aP	t1	aS	t2
		100	bb		500	100	aa
20	aa				600	100	cc
3	bb	300	Bb	20			

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ **LUNES 9:00 B** **Tiempo: 60 min** **Puntuación: 6 puntos**

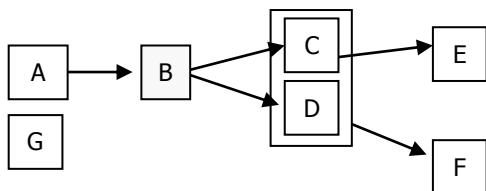
### 1) Relaciones (0/1/2 puntos)

- *milita* : los partidos pueden tener militantes
- *entrena* : los equipos pueden tener un único entrenador
- *pilota* : los pilotos pertenecen siempre a una y solo una escudería
- *pertenece* : las escuderías pueden tener uno y sólo un dueño, y los dueños en todo momento tienen una y sólo una escudería.
- *local* : los equipos siempre tienen un único estadio a su disposición, y los estadios siempre tienen un único equipo local.

Rellena, de acuerdo a las sentencias anteriores, con sus valores correspondientes las siguientes expresiones:

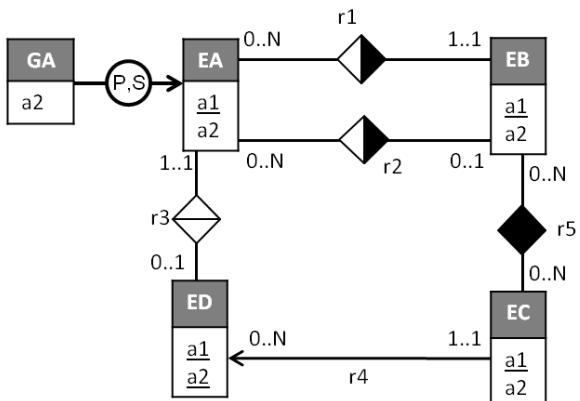
$$\begin{array}{ll}
 \text{Card(partido, milita)} = ( \textcolor{red}{0}, \textcolor{red}{N} ) & \text{Card(militante, milita)} = ( \textcolor{red}{0}, \textcolor{red}{N} ) \\
 \text{Card(eqipo, entrena)} = ( \textcolor{red}{0}, \textcolor{red}{1} ) & \text{Card(entrenador, entrena)} = ( \textcolor{red}{0}, \textcolor{red}{N} ) \\
 \text{Card(piloto, pilota)} = ( \textcolor{red}{1}, \textcolor{red}{1} ) & \text{Card(escudería, pilota)} = ( \textcolor{red}{0}, \textcolor{red}{N} ) \\
 \text{Card(dueño, pertenece)} = ( \textcolor{red}{1}, \textcolor{red}{1} ) & \text{Card(escudería, pertenece)} = ( \textcolor{red}{0}, \textcolor{red}{1} ) \\
 \text{Card(eqipo, local)} = ( \textcolor{red}{1}, \textcolor{red}{1} ) & \text{Card(estadio, local)} = ( \textcolor{red}{1}, \textcolor{red}{1} )
 \end{array}$$

### 2) Normaliza hasta 3FN o FNBC (recuerda: tablas completamente definidas) (0/1 puntos)



$R(A,G)$	$CP: (A,G)$	$CAj: A \rightarrow R1$
$R1(A,B)$	$CP: A$	$CAj: B \rightarrow R2$
$R2(B,C,D)$	$CP: B$	$CAj: (C,D) \rightarrow R3$
$R3(C,D,F)$	$CP: (C,D)$	$CAj: C \rightarrow R4$
$R4(C,E)$	$CP: C$	

**3) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)**



GA(a1,a2)	CP: a1	CAj:a1 → EA
EA(a1,a2,a3,a4)	CP: a1	CAlt: (a3,a4)      CAj: (a3,a4) → ED
ED(a1,a2)	CP: (a1,a2)	
EB(a1,a2,a3,a4)	CP: a1      VNN: a3	CAj: a3 → EA      CAj: a4 → EA
EC(a1,a2,a3,a4)	CP: (a1,a3,a4)	CAj: (a3,a4) → ED
R5(a1,a2,a3,a4)	CP: (a1,a2,a3,a4)	CAj: a1 → EB      CAj: (a2,a3,a4) → EC

**4) Dado el siguiente esquema y estado de base de datos (0/1 punto)**

**PRIMERA(p1,p2) CP(p1)**

**SEGUNDA(s1,s2,aP) CP(s1) CAj(aP)>>PRIMERA en borrados PROPAGAR modificaciones PROPAGAR**

**TERCERA(t1,aS,t2) CP(t1,aS) CAj(aS)>>SEGUNDA en borrados ANULAR modificaciones RECHAZAR**

PRIMERA		SEGUNDA			TERCERA		
p1	p2	s1	s2	aP	t1	aS	t2
1	aa	100	bb	1	500	100	aa
2	aa	200	cc	1	600	100	cc
3	bb	300	bb	2	600	200	cc

**Después de ejecutar todas las órdenes siguientes:**

delete from primera where p1=1;

update primera set p1=20 where p1=2;

delete from segunda where s1=200;

**¿cuál sería el estado final de la BD?**

PRIMERA		SEGUNDA			TERCERA		
p1	p2	s1	s2	aP	t1	aS	t2
1	aa	100	bb	1	500	100	aa
20	aa	200	cc	1	600	100	cc
3	bb	300	bb	20	600	200	cc

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_ **MIÉRCOLES 8:30**  
**Tiempo: 60 min Puntuación: 6 puntos**

**1) Modifica las siguientes tablas para reflejar las cardinalidades (0/1/2 puntos)**

**ALMACÉN**( cod, dirección, gerente, gestor)  
**CP** (cod)  
**CAlt:** (gestor)  
**CAj:** gestor → **EMPLEADO**

**ARTÍCULO**(serie, marca, modelo, fechacompra)  
**CP** (serie)

**PARTE**(tienda, número, serie, avería, fechaentrada)  
**CP**(tienda, número)  
**CAj:** tienda → **TIENDA**

**TIENDA**(cod, dirección, ciudad, provincia, administrador, teléfono)  
**CP** (cod)

**CLIENTE**(cliente, nombre, dirección, teléfono)  
**CP** (cliente)

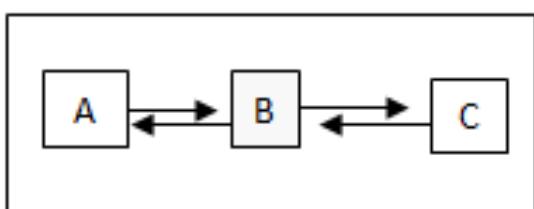
**REPARADO**(número, fecharecogida, cod)  
**CP**(número, cod)  
**CAj:** (número, cod) → **PARTE**

**LLEVAR** (serie, cliente)  
**CP** (serie, cliente)  
**CAj:** serie → **ARTICULO**  
**CAj:** cliente → **CLIENTE**

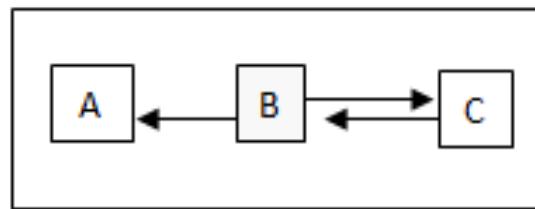
**EMPLEADO**(dni, nombre, dirección, almacén)  
**CP**(dni)  
**CAj:** almacén → **ALMACEN**

a) Card(ALMACÉN, gestiona)=(1,1)	Card(EMPLEADO, gestiona)=(0,1)
b) Card(ALMACÉN, emplea)= (0,N)	Card(EMPLEADO, emplea)=(0,1)
c) Card(CLIENTE, lleva)=(0,N)	Card(ARTÍCULO, lleva)=(0,N)
d) REPARADO es una especialización de PARTE	
e) Card(PARTE, esDe)=(1,1) (dep. identificador)	Card(TIENDA, esDe)=(0,N)

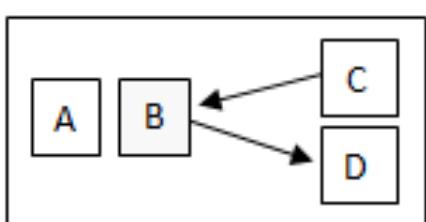
**2) En qué forma normal se encuentra cada uno de estos esquemas de dependencias funcionales (0/1 puntos)**



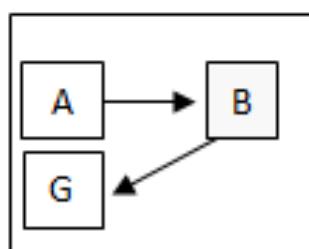
¿FN? FNBC



¿FN? FNBC



¿FN? 1FN



¿FN? 2FN

**3) Rellena el siguiente esquema con las definiciones necesarias (0/1 punto)**

**PRIMERA(p1,p2) CP(p1)**

**SEGUNDA(s1,s2,aP) CP(s1)**

**CAj(aP)>>PRIMERA**

**en borrados ANULAR/rechazar<sup>1</sup> en modificaciones PROPAGAR**

**TERCERA(t1,aS,t2) CP(t1,aS)**

**CAj(aS)>>SEGUNDA**

**en borrados RECHAZAR en modificaciones PROPAGAR**

PRIMERA	
p1	p2
1	aa
2	aa
3	bb

SEGUNDA		
s1	s2	aP
100	bb	1
200	cc	1
300	bb	2

TERCERA		
t1	aS	t2
500	100	aa
600	100	cc
600	200	cc

**Después de ejecutar todas las órdenes siguientes:**

delete from primera where p1=1;  
 update primera set p1=20 where p1=2;  
 update segunda set s1=20 where s1=200;  
 delete from segunda where s1=100;

**el estado final de la BD es el siguiente:**

PRIMERA	
p1	p2
20	aa
3	bb

SEGUNDA		
s1	s2	aP
100	bb	
20	cc	
300	bb	20

TERCERA		
t1	aS	t2
500	100	aa
600	100	cc
600	20	cc

**4) Traduce a álgebra relacional las siguientes consultas (0/1 punto)**

a) select nombre,apellidos from usuario join sitio on (usuario.sitio=sitio.cod) where sitio.cod = 3

**USUARIO x SITIO donde sitio=cod y cod=3 [nombre, apellidos]**

b) select nombre,apellidos from usuario where sitio NOT IN (select cod from cerrado)

**USUARIO[sitio] – (CERRADO[cod])  $\bowtie$  USUARIO [nombre, apellidos]**

---

<sup>1</sup> La respuesta correcta es ANULAR. Sin embargo, en el enunciado había una errata que podría inducir a que la política fuera RECHAZAR. Por ese motivo ambas respuestas se han considerado aceptables.

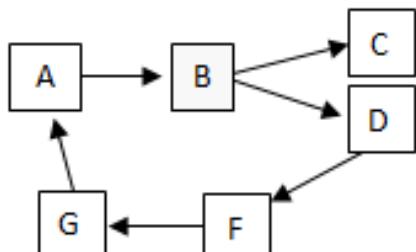
D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_ **MIÉRCOLES 10:30**  
**Tiempo: 60 min Puntuación: 6 puntos**

**1) Modifica las siguientes tablas para reflejar las cardinalidades (0/1/2 puntos)**

<b>ALMACÉN</b> ( cod, dirección, gerente, gestor) CP (cod) CAlt: gestor CAj: gestor → EMPLEADO	<b>ARTÍCULO</b> (serie, marca, modelo, fechacompra) CP (serie)	<b>PARTES</b> (número, serie, avería, fechaentrada, cod) CP(número, cod) CAj: cod → TIENDA
<b>TIENDA</b> (cod, dirección, ciudad, provincia, administrador, teléfono) CP (cod)	<b>CLIENTE</b> (cliente, nombre, dirección, teléfono) CP (cliente)	<b>REPARADO</b> (número, fecharecogida, cod) CP(número, cod) CAj: (número, cod) → PARTE
<b>LLEVAR</b> (serie, cliente) CP (serie, cliente) CAj: serie → ARTICULO CAj: cliente → CLIENTE	<b>EMPLEADO</b> (dni, nombre, dirección, almacén) CP(dni) CAj: almacén → ALMACÉN	

a) Card(ALMACÉN, gestiona)=(1,1)	Card(EMPLEADO, gestiona)=(0,1)
b) Card(ALMACÉN, emplea)= (0,N)	Card(EMPLEADO, emplea)=(0,1)
c) Card(CLIENTE, lleva)=(0,N)	Card(ARTÍCULO, lleva)=(0,N)
d) REPARADO es una especialización de PARTE	
e) Card(PARTE, esDe)=(1,1) (dep. identificador)	Card(TIENDA, esDe)=(0,N)

**2) Normaliza hasta 3FN o FNBC (0/1 puntos)**



R(A,B,C,D,F,G)

CP: A

CAlt: B

CAlt: D

CAlt: F

CAlt: G

### 3) Dado el siguiente esquema y estado de base de datos (0/1 punto)

**PRIMERA**(p1,p2) **CP(p1)**

**SEGUNDA**(s1,s2,aP) **CP(s1)** **CAj(aP)**>>**PRIMERA en borrados** PROPAGAR **modificaciones** ANULAR

**TERCERA**(t1,aP,t2) **CP(t1)** **CAlt(aP)**

**CAj(aP)**>>**PRIMERA en borrados** ANULAR **modificaciones** PROPAGAR

PRIMERA	
p1	p2
1	aa
2	aa
3	bb

SEGUNDA		
s1	s2	aP
100	bb	1
200	cc	1
300	bb	2

TERCERA		
t1	aP	t2
500	3	aa
600	1	cc
700	2	cc

**Después de ejecutar todas las órdenes siguientes:**

delete from primera where p1=1;  
update primera set p1=20 where p1=2;  
delete from segunda where s1=200;

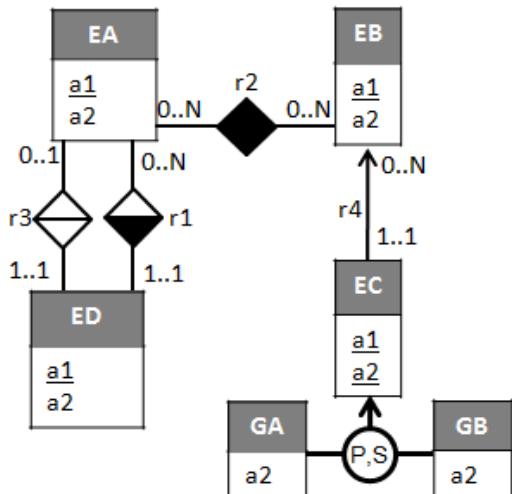
**¿Cuál sería el estado final de la BD?**

PRIMERA	
p1	p2
1	aa
20	aa
3	bb

SEGUNDA		
s1	s2	aP
100	bb	1
300	bb	

TERCERA		
t1	aP	t2
500	3	aa
600	1	cc
700	20	cc

### 4) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)



GA(a1,a2,a3,a4)	CP: (a1,a3,a4)	CAj: (a1,a3,a4) → EC
GB(a1,a2,a3,a4)	CP: (a1,a3,a4)	CAj: (a1,a3,a4) → EC
EC(a1,a2,a3)	CP: (a1,a2,a3)	CAj: a3 → EB
EB(a1,a2)	CP: a1	
EA(a1,a2)	CP: a1	
R2(a1,a2)	CP: (a1,a2)	CAj: a1 → EA      CAj: a2 → EB
ED(a1,a2,a3,a4)	CP: a1	CAlt: a3      VNN: a4      CAj: a3 → EA      CAj: a4 → EA

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_ **Tiempo: 60 min Puntuación: 6 puntos**

### 1) Relaciones (0/1/2 puntos)

- participa: los concursantes pueden participar en concursos
- preside: los comités pueden tener un único presidente
- se\_envasa: los granos de café siempre se envasan en uno y solo un bote
- rueda: el director de una película de culto siempre es uno y sólo uno, y en todo momento dirige una y sólo una película de culto.
- dirige : las cocinas siempre tienen un único chef, y los chef pueden dirigir una única cocina.

De acuerdo con las relaciones aquí descritas, empareja las expresiones de cardinalidad con sus correspondientes valores:

$$\begin{array}{ll}
 \text{Card(concursante, participa)} = ( \textcolor{red}{0} , \textcolor{red}{N} ) & \text{Card(concurso, participa)} = ( \textcolor{red}{0} , \textcolor{red}{N} ) \\
 \text{Card(comité, preside)} = ( \textcolor{red}{0} , \textcolor{red}{1} ) & \text{Card(presidente, preside)} = ( \textcolor{red}{0} , \textcolor{red}{N} ) \\
 \text{Card(grano, envasa)} = ( \textcolor{red}{1} , \textcolor{red}{1} ) & \text{Card(bote, envasa)} = ( \textcolor{red}{0} , \textcolor{red}{N} ) \\
 \text{Card(director, rueda)} = ( \textcolor{red}{1} , \textcolor{red}{1} ) & \text{Card(película, rueda)} = ( \textcolor{red}{1} , \textcolor{red}{1} ) \\
 \text{Card(cocina, dirige)} = ( \textcolor{red}{1} , \textcolor{red}{1} ) & \text{Card(chef, dirige)} = ( \textcolor{red}{0} , \textcolor{red}{1} )
 \end{array}$$

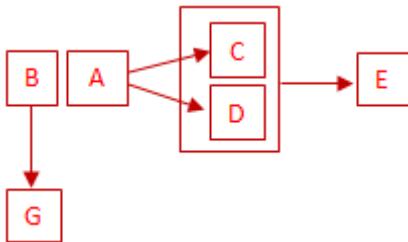
### 2) Dadas las siguientes tablas, dibuja el esquema de dependencias funcionales de donde han salido esas tablas, se entiende que normalizando desde una única tabla hasta 3FN (0/1 puntos)

**TA(a,b)**

**CP(a,b)**

**CAj(a)>>TB**

**CAj(b)>>TC**



**TC(b,g)**

**CP(b)**

**TB(a,c,d)**

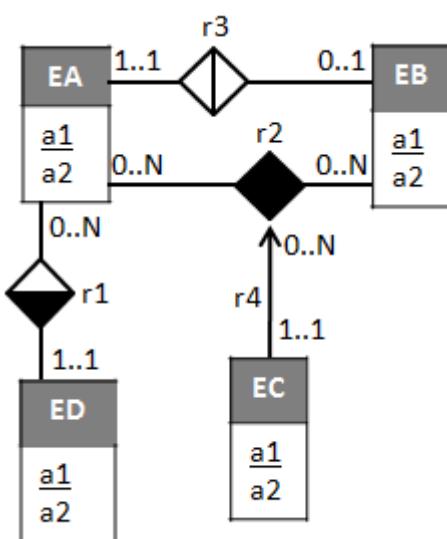
**CP(a)**

**CAj(c,d)>>TD**

**TD(c,d,e)**

**CP(c,d)**

**3) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)**



**EB (a1,a2) CP (a1)**

**EA (a1,a2,a3) CP (a1) CAj (a3)>>EB CAlt (a3)**

**ED (a1,a2,a3) CP (a1) CAj (a3)>>EA VNN (a3)**

**R2 (a1,a2) CP (a1,a2) CAj (a1)>>EA CAj (a2)>>EB**

**EC (a1,a2,a3,a4) CP (a1,a3,a4) CAj (a3,a4)>>R2**

**4) Traduce a álgebra relacional la siguiente consulta (0/1 punto)**

```

select apellidos from usuario U where
  NOT EXISTS (select 1
    from articulo A where
      NOT EXISTS (select 1
        from pedido P
        where P.dni=U.dni and P.codigo=A.codigo)
  )
  
```

PEDIDO[dni,codigo] ÷ (ARTICULO[codigo]) ⋈ USUARIO[apellidos]

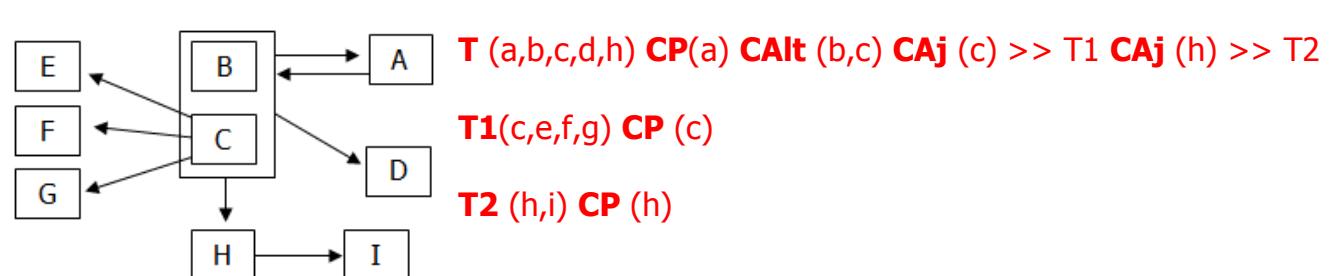
D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

**Tiempo: 60 min Puntuación: 6 puntos****1) Relaciones (0/1/2 puntos)****GRUPO** (dni:dDni, nombre: dCad)**CP(dni)****VNN (nombre)****OPERADOR** (cod:dCod, dni:dDni)**CP(cod)****CAlt (dni)****CAj (dni)** → **GRUPO****AGENTE**(cod:dCod, dni:dDni)**CP (dni)****CAlt (cod)****CAj (dni)** → **GRUPO**Card(USUARIO, realiza)= **(0,N)**Card(GRUPO, pertenece1)= **(0,1)**Card(GRUPO, pertenece2)= **(0,1)**Card(RECLAMACIÓN, atiende)= **(0,1)**Card(VISTA, es\_solucionada)= **(1,1)**Card(VISTA, soluciona)= **(0,1)****USUARIO**(cliente:dDni, nombre:dCad, dirección:dCad,

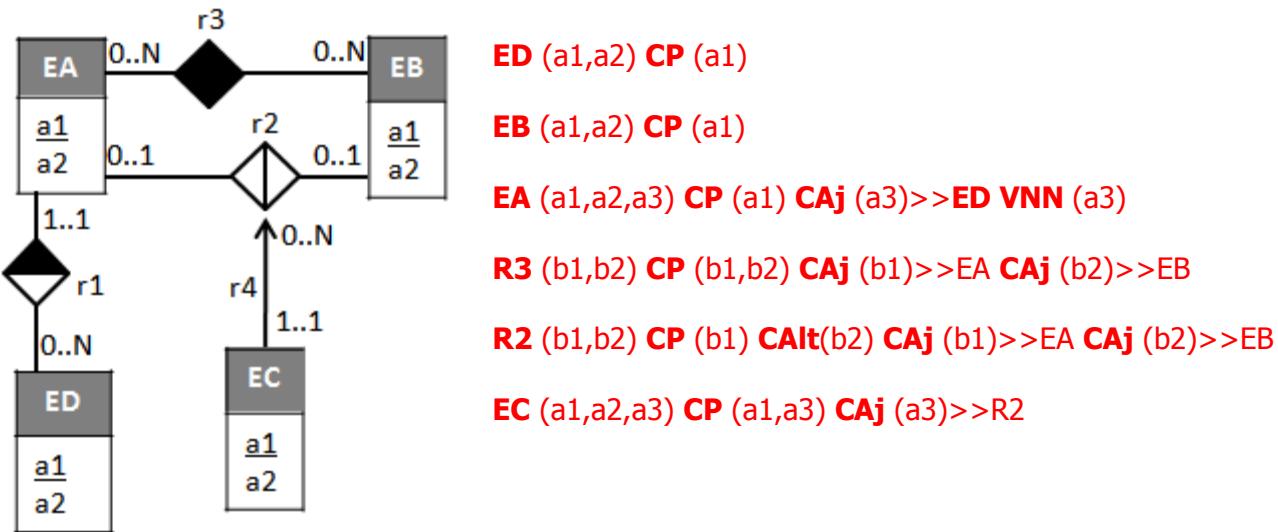
teléfono:dTlf)

**CP** (cliente)**VNN** (dirección)**VNN** (teléfono)**RECLAMACIÓN**(num:dNum, fecha:dFecha, hora:dHora,

cliente:dDni, operador:dCod)

**CP** (fecha, cliente)**CAlt** (num)**CAj** (cliente) → **USUARIO****CAj** (operador) → **OPERADOR****VISTA**(fecha:dFecha, hora:dHora, cliente:dDni, supervisor:dDni)**CP** (fecha, cliente)**CAj** (fecha, cliente) → **RECLAMACIÓN****CAj** (supervisor) → **AGENTE****2) Normaliza hasta 3FN o FNBC (0/1 puntos)**

3) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)



**ED** (a1,a2) **CP** (a1)  
**EB** (a1,a2) **CP** (a1)  
**EA** (a1,a2,a3) **CP** (a1) **CAj** (a3)>>**ED VNN** (a3)  
**R3** (b1,b2) **CP** (b1,b2) **CAj** (b1)>>**EA CAj** (b2)>>**EB**  
**R2** (b1,b2) **CP** (b1) **CAlt**(b2) **CAj** (b1)>>**EA CAj** (b2)>>**EB**  
**EC** (a1,a2,a3) **CP** (a1,a3) **CAj** (a3)>>**R2**

4) Traduce a álgebra relacional la siguiente consulta (0/1 punto)

```

select email from usuario U where
    NOT EXISTS (select 1
        from tienda A where
            NOT EXISTS (select 1
                from pedido P
                where P.dni=U.dni and P.codigo=A.codigo)
    )

```

PEDIDO[dni,codigo] ÷ (TIENDA[codigo]) ⋈ USUARIO[email]

MARCA CON UN CÍRCULO TODAS LAS RESPUESTAS QUE CREA CORRECTAS. LAS RESPUESTAS INCORRECTAS RESTAN UNA CORRECTA. LAS RESPUESTAS NO ELEGIDAS NO PENALIZAN.

El entidad-relación

- a) es el modelo de datos subyacente en el motor de MySQL, Oracle y SQL Server.
- b) **es semántico y el modelo relacional es clásico.**
- c) es semántico, al igual que el modelo relacional.
- d) no tiene traducción posible a modelo relacional

La integridad de clave

- a) se aplica a la clave primaria pero no a la clave alternativa
- b) exige que la clave ajena sea complemento nula o contenga un valor previamente almacenado en una clave primaria
- c) **en ninguna clave candidata permite nulos**

Una tabla en el modelo relacional

- a) **sólo puede estar en tercera forma normal si lo está en primera y en segunda**
- b) puede estar en tercera forma normal aunque no esté en segunda
- c) puede estar en tercera forma normal aunque no esté en primera

El concepto de relación matemática se adapta al modelo relacional

- a) **asignando un nombre simbólico a los componentes de las tuplas de la relación**
- b) creando los conceptos de clave alternativa y clave ajena
- c) permitiendo que las tuplas puedan repetirse y que las relaciones no estén obligadas a tener clave primaria

La organización indexada se basa en la idea de que

- a) **es más fácil manejar archivos pequeños que incluso podrían caber en RAM para encontrar un registro concreto**
- b) la selección de un campo cualquiera para la ordenación de listados de datos es necesaria solo en muy contados casos.
- c) no se necesita insertar con demasiada frecuencia
- d) **aunque se ocupa más espacio en disco su eficiencia en determinadas operaciones la aconseja**

Es cierto que

- a) **para ciertos problemas o aplicaciones es mejor no utilizar SGBD**
- b) cualquier SGBD relacional sólo puede manejar BD normalizadas como mínimo hasta 3FN (o FNBC si hiciera falta)
- c) el modelo entidad-relación es mejor que el modelo relacional

En E-R, la generalización

- a) puede ser al mismo tiempo parcial, total, disjunta y solapada.
- b) **es una representación de subtipos de una clase de objetos general.**
- c) **se define siempre con propiedades de cobertura.**
- d) **si es parcial no es total, y si es disjunta no es solapada.**

La integridad referencial en un SGBD relacional

- a) **se cumple si toda la clave ajena es nula**
- b) es la restricción que garantiza la no duplicidad de tuplas.
- c) **se cumple si ningún atributo de la clave es nulo y la referencia es válida.**

Para evitar redundancias en una base de datos relacional

- a) la política adecuada para mantener la integridad referencial es la de anular, aunque a veces es decisión del diseñador emplear otra política
- b) **se aplica el proceso de normalización hasta obtener relaciones en 3FN (en la mayoría de los casos)**
- c) las claves ajenas deben tener prevista la política adecuada ante operaciones de borrado

El modelo relacional NO fue desarrollado teóricamente por

- a) Codd
- b) **Navathe**
- c) Chen

Al efectuar un producto cartesiano de álgebra relacional de una tabla por si misma

- a) necesitamos definir alias de relación.
- b) las columnas de la tabla derivada no tienen nombre.
- c) por definición, nunca produce duplicados de tuplas.

En cuanto a los esquemas resultado de las distintas fases de desarrollo de software

- a) el esquema conceptual se genera teniendo en cuenta que se va trabajar en una base de datos.
- b) el esquema conceptual depende de una máquina y software concreta.
- c) el esquema físico es lo más cercano al disco duro, al mantenimiento de ficheros.
- d) ninguno depende de otro, el lógico no depende del conceptual, el físico no depende del conceptual, etc.

Son notaciones distintas del E-R

- a) atributos compuestos.
- b) claves ajenas.
- c) IDEF1X
- d) ficheros indexados.

El nombre de las columnas del resultado de realizar una concatenación natural en álgebra relacional

- a) son los mismos que los de la segunda relación.
- b) son todos los de la primera relación más los no comunes de la segunda.
- c) son los mismos que los de la primera relación.

Fueron precursores del cálculo relacional lenguajes como

- a) Cobol.
- b) SQL.
- c) ALPHA y QBE (Query By Example).

La dependencia funcional es

- a) la cualidad de las columnas de una tabla que solo admiten valores escalares, valores simples
- b) una restricción del modelo que dice que las claves candidatas no admiten nulos, ni total ni parcialmente
- c) una relación entre atributos de una tabla de tal forma que un valor de uno determina únicamente el valor del otro

La crisis del software, entre otras, tenía las siguientes características

- a) las empresas no pagaban por el software que encargaban
- b) los proyectos eran demasiado pequeños
- c) el software a menudo no cumplía con los requerimientos
- d) los programadores no trabajaban lo suficiente

La arquitectura a varios niveles de un SGBD

- a) se abandonó con la llegada del modelo relacional.
- b) permite definir varios niveles de restricciones que se corresponden con clave primaria, alternativa, ajena y valor no nulo.
- c) está muy relacionada con el objetivo de independencia de datos que debe lograr un SGBD.

El esquema es

- a) una descripción de un lenguaje concreto.
- b) el resultado de aplicar un determinado modelo de datos.
- c) una descripción de ficheros.
- d) un lenguaje de programación como C, PHP o Java.

Ventajas de la arquitectura cliente-servidor

- a) los datos están en el servidor y es el cliente el que da formato a esos datos para presentarlos en pantalla
- b) los datos están en el cliente y es el servidor el que da formato a esos datos para presentarlos en pantalla
- c) centraliza grandes bases de datos en una sola máquina servidor, eliminando toda redundancia posible.

## 1) Relaciones (0/1/2 puntos)

NOTA: aunque no se muestran, los dominios se suponen bien definidos.

**EVENTO** (cod, descripción) **CP** (cod)

**DISTRITO** (num) **CP** (num)

**CUADRANTE** (num, cuad)

**CP** (num, cuad)

**CAj** (num) >> DISTRITO

**UNIDAD** (nuni,nombre, num,cuad) **CP** (nuni) **CAlt** (nombre)

**CAj** (num,cuad) >> CUADRANTE

**VNN**(num)

**VNN**(cuad)

**LOCALIZACIÓN** (num,unidad,extensión,tipo) **CP**(num)

**CAj** (unidad) >> UNIDAD

Card(LOCALIZACIÓN, tiene)= **(0,N)**

Card(EVENTO, por)= **(0,N)**

Card(CATEGORÍA1, padre)= **(0,N)**

Card(UNIDAD, vigila)= **(1,1)**

Card(UNIDAD, sita)= **(0,N)**

Card(UNIDAD, se\_especializa)= **(0,N)**

Card(UNIDAD, actuación)= **(0,1)**

**SALIDA** (sal, línea, cod, orientación) **CP** (sal, línea)

**CAj** (sal) >> LOCALIZACIÓN

**CAj** (cod) >> EVENTO

**CATEGORÍA** (concepto, relacionado) **CP** (concepto)

**CAj** (relacionado) >> CATEGORÍA

**ACTUACIÓN** (nuni, cat) **CP** (nuni) **CAlt** (cat)

**CAj** (nuni) → UNIDAD

**CAj** (cat) → CATEGORÍA

**ESPECIALIZADA** (nuni, cat) **CP** (nuni, cat)

**CAj** (nuni) → UNIDAD

**CAj** (cat) → CATEGORÍA

Card(SALIDA, tiene)= **(1,1)**

Card(SALIDA, por)= **(0,1)**

Card(CATEGORÍA2, padre)= **(0,1)** \*

Card(CUADRANTE, vigila)= **(0,N)**

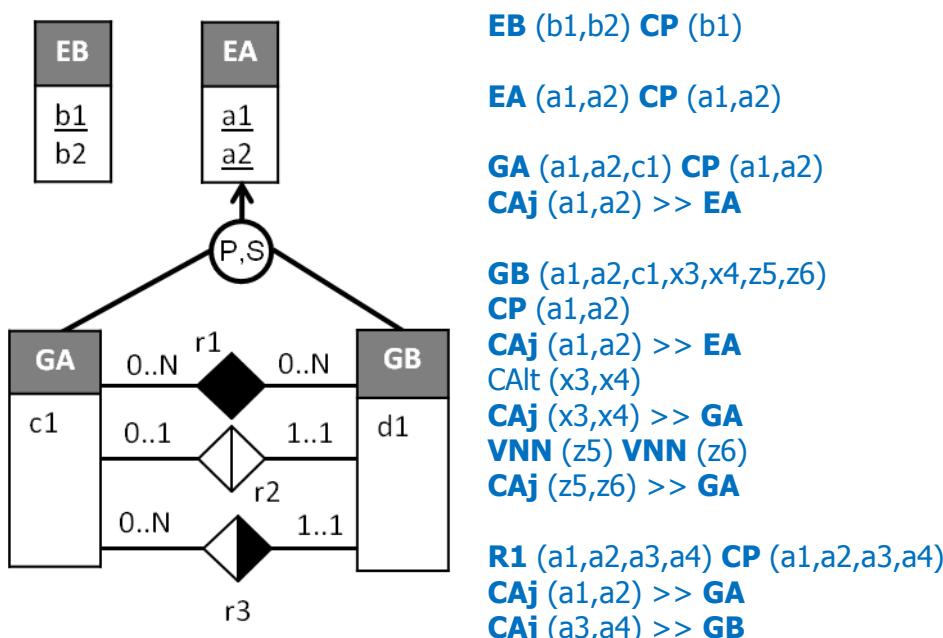
Card(LOCALIZACIÓN, sita)= **(0,1)**

Card(CATEGORÍA, se\_especializa)= **(0,N)**

Card(CATEGORÍA, actuación)= **(0,1)**

\* siendo una relación reflexiva, los valores pueden ponerse de forma inversa, **(0,1)** **(0,N)**

## 2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2 puntos)



- 3) Cuál será el siguiente estado de base de datos si se ejecutaran consecutivamente las siguientes órdenes (0 o 1):

UNIVERSIDAD (NUM,NOMBRE)  
CP(NUM)

DEPARTAMENTO (COD,U)  
CP(COD, U)  
CAJ (U) >> (UNIVERSIDAD)  
BOR RECHAZAR MOD RECHAZAR

PROFESOR (DNI,NOMBRE ,DPTO,U)  
CP(DNI),  
CAJ(DPTO,U)>> (DEPARTAMENTO)  
BOR PROPAGAR MOD PROPAGAR

UNIVERSIDAD	
NUM	NOMBRE
1	ALICANTE
2	VALENCIA
3	JAUME I
4	CARLOS III

PROFESOR		DEPARTAMENTO			
DNI	NOMBRE	DPTO	U	COD	U
21	RENATA	DLSI	1	DLSI	1
22	FLOR	DLSI	2	DLSI	2
23	ABUNDIO	DLSI	1	DLSI	3
13	SEGISMUNDO	CCIA	1	CCIA	1
31	YENIPHER	CCIA	1		

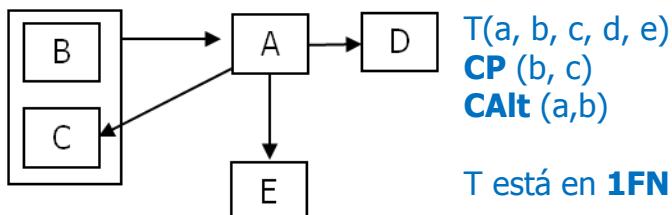
Delete from profesor where nombre='FLOR';  
delete from departamento where cod='DLSI' and U=3;  
update universidad set num=5 where num=4;

PROFESOR			
DNI	NOMBRE	DPTO	U
21	RENATA	DLSI	1
23	ABUNDIO	DLSI	1
13	SEGISMUNDO	CCIA	1
31	YENIPHER	CCIA	1

DEPARTAMENTO	
COD	U
DLSI	1
DLSI	2
CCIA	1

UNIVERSIDAD	
NUM	NOMBRE
1	ALICANTE
2	VALENCIA
3	JAUME I
5	CARLOS III

- 4) ¿En qué forma normal se encuentra la tabla resultado de aplicar estas dependencias funcionales (0/1 puntos)? Define esa tabla inicial e indica la forma normal en que se encuentra.



T(a, b, c, d, e)  
CP (b, c)  
CAlt (a,b)

T está en 1FN

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

Aula y fecha: \_\_\_\_\_

**Tiempo: 2 horas****Puntuación: 9 (x 5.556 = 50 puntos)****1) Relaciones (0/1/2=más de 2 errores / 2 errores / 1 o sin errores)****PARTIDO**(partido, nombre)**CP**(partido)**MILITANTE**(dni, nombre, milita)**CP**(dni)**CAj**(milita)>>PARTIDO**POSICIÓN**(cargo, militante, partido)**CP**(cargo, militante)**CAj**(cargo)>>CARGO**CAj**(militante)>>MILITANTE**CAj**(partido)>>PARTIDO**VNN**(partido)**AMIGOTES**(dni, nombre, partido)**CP**(dni)**CAlt**(nombre) **CAj**(partido)>>PARTIDO**BANQUEROS**(dni) **CP**(dni)**CAj**(dni)>>AMIGOTES**CARGO**(cargo)**CP**(cargo)

Rellena, de acuerdo a las sentencias anteriores, con sus valores correspondientes las siguientes expresiones:

Card(partido, milita) = (0,N)

Card(militante, milita) = (0,1)

Card(cargo, posición) = (0,N)

Card(militante, posición) = (0,N)

Card(posición, en\_el) = (1,1)

Card(partido, en\_el) = (0,N)

Card(amigotes, amigos) = (0,1)

Card(partido, amigos) = (0,N)

Card(banquero, es) = (1,1)

Card(amigote, es) = (0,1)

**2) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna / 1 correcta / 2 correctas)**

- a) DNI y nombre de los militantes tales que su DNI también aparece en amigotes pero con distinto nombre.

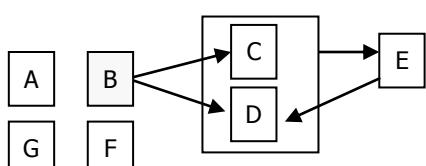
**MILITANTE x AMIGOTES**donde **MILITANTE.dni** = **AMIGOTES.dni** y **MILITANTE.nombre** ≠ **AMIGOTES.nombre**[**MILITANTE.dni**, **MILITANTE.nombre**]

- b) Nombre del militante (o militantes) que es presidente (el cargo es "presidente") de todos los partidos conocidos.

**POSICIÓN** donde cargo="presidente" [**militante,partido**]

÷

(PARTIDO[partido])

× **MILITANTE** donde militante=dni [nombre]**3) Normaliza hasta 3FN o FNBC (recuerda: tablas completamente definidas) (0/1/2=más de 1 error / 1 error / sin errores)****T1 (A, B, F, G)****CP(A, B, G, F)****CAj(B)>>T2****T2 (B, C, E)****CP(B)****CAj(C,E)>>T3****T3 (C, E)****CP(C,E)****CAj(E)>>T4****T4 (E, D)****CP(E)**

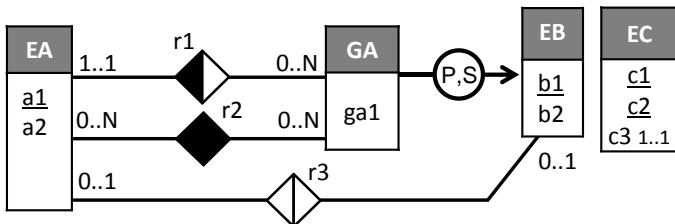
En una primera etapa, la normalización llega a las siguientes tablas:

**T1 (A, B, F, G)****CP(A, B, G, F)****CAj(B)>>T2****T2 (B, C, D, E)****CP(B)****CAj(C,D)>>T3****T3 (C, D, E)****CP(C, D)****CAlt(C, E)**T1 y T2 están en FNBC, no hay nada más que hacer en ellas. Sin embargo, **T3 está en 3FN pero no en FNBC**: el determinante (E) no es clave candidata. Esto se arregla igual que todo lo demás, (E,D) van a otra tabla y D se elimina de la original.

Es precisamente la eliminación de esta columna (D) de T3 la que complica todo y hay que hacer arreglos tanto en T3 como en T2. En primera instancia, la clave primaria de T3 ya no puede serlo, D nos la hemos llevado a T4. Pero si cambiamos esta CP, entonces la T2 queda inconsistente, nos obliga a sustituir D por E.

**T1 (A, B, F, G)****CP(A, B, G, F)****CAj(B)>>T2****T2 (B, C, D, E)****CP(B)****CAj(C,D)>>T3****T3 (C, D, E)****CP(C, D)****CAlt(C, E)****CAj(E)>>T4****T4 (E, D)****CP(E)**

4) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores)



**TC** (c1, c2, c3)      **R2** (a1, b1)  
**CP**(c1,c2)      **CP(a1, b1)**  
**VNN(c3)**      **CAj(a1)>>TA**  
**CAj(b1)>>TGA**

**TB** (b1, b2)      **R3** (a1, b1)  
**CP(b1)**      **CP(a1)**  
**TGA** (b1, ga1)      **CAlt(b1)**  
**CP(b1)**      **CAj(a1)>>TA**  
**CAj(b1)>>TB**      **CAj(b1)>>TB**

**TA** (a1, a2, b1)  
**CP(a1)**  
**CAj(b1)>>TGA**  
**VNN(b1)**

5) Un amigo nuestro dice que es capaz de definir una relación muchos a muchos con doble restricción de existencia

$\text{Card}(T1,r)=(1,N)$

$\text{Card}(T2,r)=(1,N)$

con el siguiente esquema:

$T1(a,b) \text{ CP}(a) \text{ CAj}(b)>>T2 \text{ VNN}(b)$

$T2(c,d) \text{ CP}(c) \text{ CAj}(d)>>T1 \text{ VNN}(d)$

Suponiendo que los dominios están correctamente definidos aunque no se muestran, y suponiendo también que la doble restricción de existencia se ha previsto y no impide insertar filas en cualquiera de las tablas, ¿cuál es el fallo en el que ha incurrido nuestro amigo? (0/1=mal / bien)

Aunque para insertar una fila en T1 debe haber previamente otra en T2 para poder relacionarlas, y lo mismo ocurre en T2 respecto de T1, el enunciado nos dice que no nos preocupemos de eso, que lo importante es que se ha conseguido en el esquema de base de datos reflejar esas cardinalidades mínimas igual a 1.

El problema real es que hemos perdido la relación muchos a muchos. Sea el siguiente estado de base de datos E1: da igual los valores almacenados en las claves ajenas, lo importante es que solo podemos hacer 6 asociaciones entre filas de T1 y T2

Pero en una relación muchos a muchos es posible relacionar a todos con todos, como se ve en estado de la base de datos E2: con las 3 filas de T1 y las 3 de T2, hasta 9 combinaciones.

Digamos que se da preferencia a las cardinalidades máximas sobre las mínimas, es más importante disponer de una relación muchos a muchos que establecer dobles restricciones de existencia que, por otra parte, son difíciles de manejar y de justificar. Por eso, las relaciones N:M se representan siempre con una tercera tabla con dos claves ajenas y decimos que es "imposible" imponer restricciones de existencia a ninguna de las tablas a relacionar.

Además, 2 claves ajenas, cada una en una tabla diferente, almacenan sus valores de forma autónoma, no hay manera de coordinarlas (salvo que se escriba un programa en una capa lógica superior): es que son dos relaciones independientes, es decir, que se caracterizan como:

$\text{Card}(T1, r) = (1,1)$   $\text{Card}(T2, r)=(0,N)$

$\text{Card}(T1, s) = (0,N)$   $\text{Card}(T2, s)=(1,1)$

a	b
1	x
2	x
3	x

c	d
x	1
y	1
z	1

Estado de la base de datos E1

a	c	aT1	aT2
1	x	1	x
2	y	1	y
3	z	1	z
		2	x
		2	y
		2	z
		3	x
		3	y
		3	z

Estado de la base de datos E2



11 de julio de 2012 FBD TEST (50 PUNTOS) M1 DNI: \_\_\_\_\_

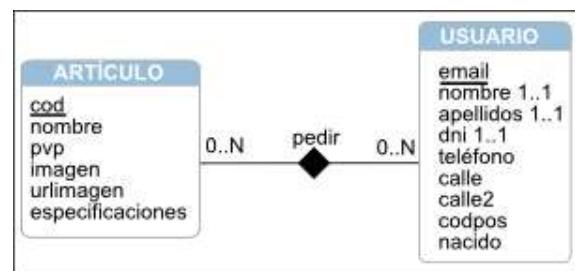
NOMBRE: \_\_\_\_\_

MARCA CON UN CÍRCULO TODAS LAS RESPUESTAS QUE CREAIS CORRECTAS. LAS RESPUESTAS INCORRECTAS RESTAN UNA CORRECTA. LAS RESPUESTAS NO ELEGIDAS NO PENALIZAN.

- 1) Los problemas detectados en los sistemas de ficheros antes de la llegada de las técnicas de bases de datos se pueden resumir en
  - a) redundancia de datos
  - b) seguridad insuficiente
  - c) los programas dependen de los datos
  - d) los programas dependen de datos que no usan
- 2) La tecnología magnética aplicada al almacenamiento masivo de datos se aplica a
  - a) salvo a las tarjetas perforadas, a todos
  - b) discos duros y flexibles
  - c) CD, DVD y Blu Ray
  - d) discos duros, discos flexibles y discos flash
- 3) Un índice denso
  - a) es un índice con muchos datos
  - b) requiere que el fichero de datos esté ordenado por un campo clave
  - c) estructuran los registros en forma de árbol
  - d) no pueden ser multinivel
- 4) La organización secuencial ordenada
  - a) es la peor para obtener un listado ordenado por el campo clave.
  - b) es la mejor para obtener un listado ordenado por cualquier campo.
  - c) obliga a una reorganización del fichero cada vez que quiero insertar un registro intermedio.
  - d) optimiza el espacio
- 5) Si tengo dos tablas definidas como
 
$$A(a,b) \text{ CP}(a) CAj(b) \rightarrow B VNN(a)$$

$$B(c,d) \text{ CP}(c) CAj(d) \rightarrow A VNN(d)$$
  - a) B.d no puede ser clave ajena porque la clave primaria a la que apunta se llama "a", A.a.
  - b) se trata de una relación uno a uno, porque la restricción de existencia independiente ( $Card(A,r)=1..1$ ) ( $Card(B,s)=1..1$ ) ( $Card(A,r)=1..1$ ) ( $Card(B,s)=1..1$ ).
  - c) tenemos 2 relaciones 1:N con restricción de existencia independientes:  $Card(A,r)=0..N$ ,  $Card(B,r)=1..1$  y  $Card(A,s)=1..1$ ,  $Card(B,s)=0..N$ .
- 6) Un modelo de datos es
  - a) todo aquello que se pueda consultar con una orden select.
  - b) una base de datos ejemplo para todas las demás.
  - c) una teoría de especificación describiendo como una base de datos se estructura y se usa.
  - d) un esquema de base datos como, por ejemplo, el de TiendaOnLine.
- 7) Según este esquema E-R

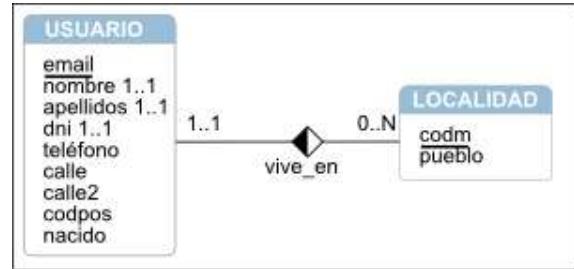
- a) los usuarios solo pueden pedir N artículos, donde N es un valor indeterminado entre 0 y 1.
- b) los usuarios pueden pedir todos los artículos existentes, si quieren.
- c) los usuarios siempre tienen email, dni, nombre y apellidos.
- d) todos los usuarios deben haber pedido algo.



- 8) select T1.a,T1.b,T2.c from T1, T2 where T1.b=T2.b, en álgebra relacional, podría ser
  - a) la concatenación natural.
  - b) el producto cartesiano.
  - c) la unión.
  - d) la división.
- 9) QBE es una implementación de
  - a) cálculo relacional de tuplas
  - b) cálculo relacional de dominios
  - c) álgebra relacional



- 10) Una consulta en álgebra relacional que sea parecida a "dame TODOS los vendedores de la provincia de Alicante"
- no se puede hacer porque precisa hacer la cuenta de filas que hay en la tabla, y el álgebra relacional está muy limitado para contar, ordenar, etc.
  - no necesita el operador DIVISIÓN**
  - necesita el operador DIVISIÓN.
- 11) Un defecto de normalización en una base de datos relacional puede provocar anomalías
- al borrar la información de una tupla, ya que se pueden estar borrando tuplas de otras tablas involuntariamente
  - al insertar información en una tabla, porque los datos insertados no se corresponderán con la realidad
  - al modificar la información de una tabla, ya que un cambio simple de un dato podría afectar a varias tuplas.**
- 12) Si en una tabla no hay dependencias funcionales entre sus atributos
- habrá tantas claves candidatas como columnas tenga la tabla
  - esa tabla solo tiene una clave candidata**
  - eso nunca ocurrirá en modelo relacional: toda tabla tiene al menos una dependencia funcional
- 13) La normalización
- se utiliza actualmente más como un criterio de calidad en el diseño**
  - no es necesaria aunque sí recomendable
  - es el mejor método para diseñar las bases de datos
- 14) De la arquitectura cliente-servidor
- es una nueva arquitectura de SGBD basada en otros esquemas
  - es un esquema más de la arquitectura de un SGBD
  - es una organización distribuida donde el SGBD, si lo hay, haría el papel de servidor**
- 15) Según este esquema E-R
- a) los usuarios pueden tener varias localidades de residencia, la principal y la de veraneo, por ejemplo.  
b) los usuarios se identifican por su DNI.  
**c) los usuarios siempre viven en al menos una localidad.**  
d) las localidades pueden NO tener habitantes.
- 16) El trabajar con un SGBD que siga el Modelo Relacional fielmente nos garantiza
- que en las tablas no hay tuplas duplicadas.**
  - la posibilidad de utilizar columnas multivaluadas.
  - que no hay redundancia de información.
- 17) Si como producto de la adaptación del concepto de relación matemática al modelo relacional decimos que la relación tiene intención y extensión, la segunda se define como
- el conjunto de n-tuplas, donde cada tupla es un conjunto de pares (nombreAtributo: valor)**
  - un conjunto de dominios no necesariamente disjuntos
  - un conjunto de nombres de atributos distintos, cada uno de ellos asociado a su dominio correspondiente
- 18) Una clave ajena en el modelo relacional
- indica una asociación entre objetos.**
  - es un tipo de clave candidata.
  - nunca podrá estar formada por todos los atributos de una relación.
- 19) Al hablar de claves en el Modelo Relacional
- una clave candidata de una relación siempre debe ser también clave ajena.
  - una clave ajena de una relación R puede ser también la clave primaria de R.**
  - una clave ajena de una relación R debe coincidir en el número de columnas con el de la clave primaria de R.
- 20) En cuanto a las distintas notaciones del E-R
- no es cierto, solo hay una única notación.
  - unas sirven para el esquema conceptual, otras para el lógico y otras para el físico.
  - nos referimos al nombre que le daremos al esquema de base de datos.
  - nos da igual, podremos obtener un esquema equivalente en cada una de ellas.**



D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

Aula y fecha: \_\_\_\_\_

**Tiempo: 2 horas****Puntuación: 10** ( $\times 5 = 50$  puntos)

- 1) Modifica el esquema de la base de datos para contemplar las siguientes correspondencias y cardinalidades entre las relaciones (0/1/2=más de 2 errores / 2 errores / 1 o sin errores)**

**PARTIDO**(partido, nombre)**CP**(partido)**CARGO**(cargo)**CP**(cargo)**AMIGOTES**(dni, nombre)**CP**(dni)**CAlt**(nombre)**MILITANTE**(dni, nombre)**CP**(dni)**BANQUEROS**(dni)**CP**(dni)

Card(partido, milita) = (0,N)

Card(cargo, posición) = (0,1)

Card(posición, en\_el) = (1,1)

Card(amigotes, amigos) = (0,N)

Card(banquero, es) = (1,1)

Card(militante, milita) = (1,1)

Card(militante, posición) = (0,1)

Card(partido, en\_el) = (0,N)

Card(partido, amigos) = (0,1)

Card(amigote, es) = (0,1)

**PARTIDO**(partido, nombre,  
amigote)**CP**(partido)**CAj**(amigote)>>**AMIGOTES****POSICIÓN**(cargo, militante, partido)**CP**(cargo)**CAlt**(militante)**AMIGOTES**(dni, nombre)**CP**(dni)**CAlt**(nombre)**MILITANTE**(dni, nombre, milita)**CP**(dni)**CAj**(milita)>>**PARTIDO****VNN**(milita)**CAj**(cargo)>>**CARGO****CAj**(militante)>>**MILITANTE****CAj**(partido)>>**PARTIDO****VNN**(partido)**BANQUEROS**(dni)**CP**(dni)**CAj**(dni)>>**AMIGOTES****CARGO**(cargo)**CP**(cargo)*Alternativamente, sería equivalente:***BANQUEROS**(dni, amigote)**CP**(dni)**CAlt**(amigote)**CAj**(amigote)>>**AMIGOTES**

- 2) Dado el esquema de base de datos del enunciado anterior, y suponiendo que todos los dominios son iguales, proporciona enunciados válidos para las siguientes expresiones en álgebra relacional (0/1/2=ninguna / 1 correcta / 2 correctas)**

- a) Definir alias A1 para AMIGOTES

Definir alias A2 para AMIGOTES

AMIGOTES[dni] - (A1 x A2 donde A1.nombre &gt; A2.nombre [A1.dni])

Dni del amigote cuyo nombre es el primero alfabéticamente.

- b) MILITANTE [dni,nombre] ÷ (AMIGOTES [nombre])

Dni del militante (o militantes) que se llama igual que todos los amigotes



- 3) Transforma este esquema relacional en un esquema entidad-relación equivalente (0/1/2=más de 1 error / 1 error / sin errores)

**TA** (a1, a2, a3)

**CP(a1)**

**CAlt(a2)**

**TB** (b1, b2, b3)

**CP(b1, b2, b3)**

**CAj(b1)>>TA**

**TC** (c1, c2, c3)

**CP(c1)**

**TD(d1,d2,d3)**

**CP(d1)**

**CAj(d1)>>TC**

**VNN(d2)**

**TE (e1)**

**CP(e1)**

**CAj(e1)>>TC**

**TF (f1,f2,f3)**

**CP(f1)**

**TG (g1, g2)**

**CP(g1)**

**CAlt(g2)**

**CAj(g1)>>TA**

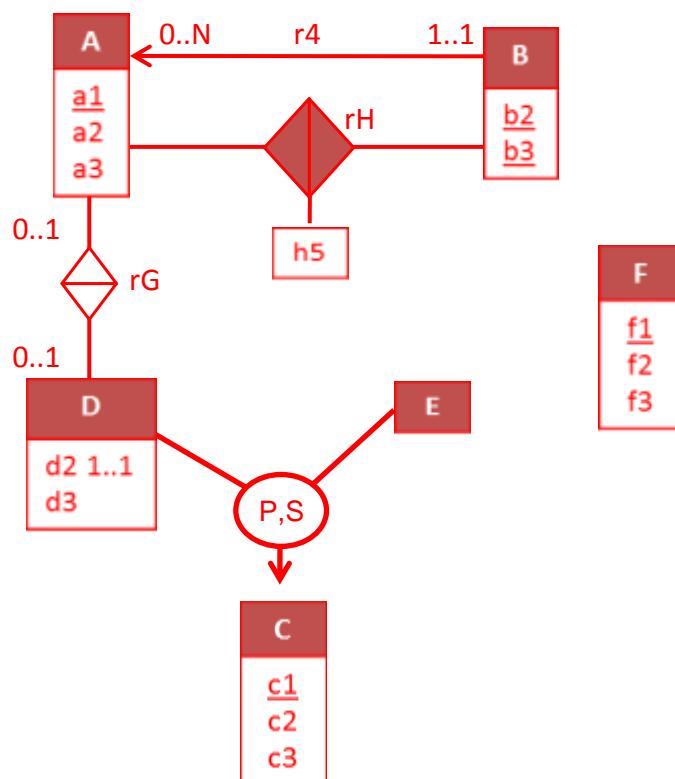
**CAj(g2)>>TD**

**TH(h1,h2,h3,h4,h5)**

**CP(h1,h2,h3,h4)**

**CAj(h1)>>TA**

**CAj(h2,h3,h4)>>TB**



**4) Suponiendo todos los dominios iguales, localiza los 5 errores que ha cometido el diseñador de esta base de datos (0/1/2=faltan más de 2 errores por localizar/ faltan 2 errores / falta 1 o ningún error)**

**T1** (A, B, C, D)

**CP**(A, B, C)

**CAlt**(B,C)

**CAj**(B,C)>>T2

(propagar modificaciones,  
rechazar borrados)

**VNN**(B)

**T3** (A, B, C, D, E)

**CP**(A, B, C, D)

**CAj**(B,C)>>T1

(rechazar modificaciones,  
rechazar borrados)

**CAj**(C,E)>>T2

(rechazar modificaciones,  
anular borrados)

1) En T1, la CAlt es subconjunto de CP

2) En T1, el VNN es innecesario al ser B parte de la CP

3) En T3 CAj(B,C)>>T1 apunta a una CP formada por tres columnas

4) En T3, la política anular borrados en CAj(C,D)>>T2 no puede aplicarse al ser C parte de la CP

5) En T4 falta la CP

**T2** (C, D, E)

**CP**(C,D)

**CAlt**(E)

**CAj**(D,E)>>T2

(propagar modificaciones,  
propagar borrados)

**T4** (C, D, E)

**CAlt** (E)

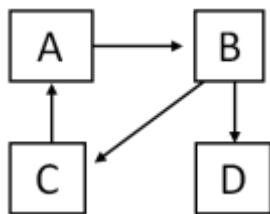
**CAj**(C,D)>>T2

(rechazar modificaciones,  
anular borrados)

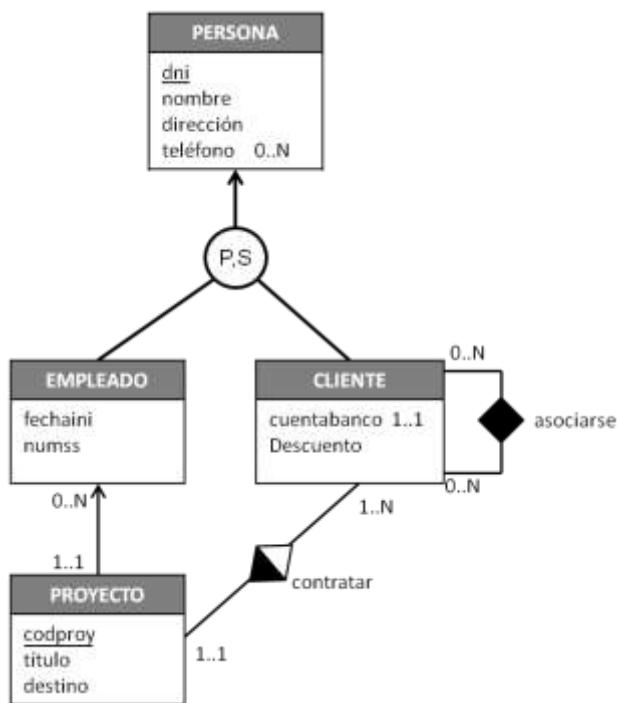


D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

Aula y fecha: \_\_\_\_\_ Tiempo: 1 hora 15 min Puntuación: 10 (x4 = 40 puntos)

**1) Normaliza hasta FNBC (recuerda: tablas completamente definidas excepto dominios) (0/2)**

**T(a,b,c,d)**  
**CP(a)**  
**CAlt(b)**  
**CAlt(c)**

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.**

**PERSONA(dni, nombre, dirección, teléfono)\***  
**CP(dni)**  
**EMPLEADO(dni,fechaini,numss)**  
**CP(dni)**  
**CAj(dni)>>PERSONA**  
**CLIENTE(dni, cuentabanco, descuento)**  
**CP(dni)**  
**CAj(dni)>>PERSONA**  
**VNN(cuentabanco)**  
**ASOCIARSE(cli1, cli2)**  
**CP(cli1,cli2)**  
**CAj(cli1)>>CLIENTE**  
**CAj(cli2)>>CLIENTE**  
**PROYECTO(empleado, codprop, título, destino, cliente)\*\***  
**CP(empleado, codprop)**  
**CAj(empleado)>>EMPLEADO**  
**CAj(cliente)>>CLIENTE**

**VNN(cliente)****\* se pierde el atributo multivaluado (aunque se podría haber solucionado con otra tabla)****\*\* se pierde la restricción de existencia de CLIENTE contra CONTRATAR**

### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de exámenes de tipo cuestionario con preguntas de varias opciones. Cada examen está compuesto de un conjunto de preguntas (tabla CONTENIDO) y las respuestas que los estudiantes consideran correctas se almacenan en RESPUESTAS.

PREGUNTA(codpr,enunciado)	EXAMEN(num,nombre,puntua)	RESPUESTAS(es,exa,codpr,opcion)
CP(codpr)	CP(num)	CP(es,exa,codpr,opcion)
POPCIONES(codpr,opcion,texto)	CONTENIDO(exa,codpr,puntua)	CAj(es)>>ESTUDIANTE
CP(codpr,opcion)	CP(exa,codpr)	CAj(exa)>>EXAMEN
CAj(codpr)>>PREGUNTA	CAj(exa)>>EXAMEN	CAj(codpr,opcion)>>POPCIONES
CORRECTAS(codpr,opcion)	CAj(codpr)>>PREGUNTA	CAj(exa,codpr)>>CONTENIDO
CP(codpr,opcion)	ESTUDIANTE(dni,nombre)	
CAj(codpr,opcion)>> POPCIONES	CP(dni)	

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                       |                                        |
|---------------------------------------|----------------------------------------|
| a) Card(PREGUNTA, tiene) = ( 0, N )   | b) Card(POPCIONES, tiene) = ( 1, 1 )   |
| c) Card(POPCIONES, son) = ( 0, 1 )    | d) Card(CORRECTAS, son) = ( 1, 1 )     |
| e) Card(EXAMEN, contiene) = ( 0, N )  | f) Card(PREGUNTA, contiene) = ( 0, N ) |
| g) Card(ESTUDIANTE, hace) = ( 0, N )  | h) Card(EXAMEN, hace) = ( 0, N )       |
| i) Card(RESPUESTAS, marca) = ( 1, 1 ) | j) Card(POPCIONES, marca) = ( 0, N )   |

### 4) Responde con verdadero o falso (0/1/2=3 o más errores/2/hasta 1 error; no contestada=error)

- |   |
|---|
| F |
| F |
| F |
| V |
| F |
| F |
| F |
- a) en cada examen, el estudiante debe responder\* a todas las preguntas, no puede dejar preguntas sin contestar
  - b) un estudiante no puede responder\* a la misma pregunta varias veces si es en exámenes distintos
  - c) en RESPUESTAS, la clave ajena (exa,codpr)>>CONTENIDO está mal, no debiera estar definida ahí
  - d) en RESPUESTAS, la clave ajena (exa,codpr)>>CONTENIDO nos asegura que no se responde a opciones de preguntas que no pertenecen al examen
  - e) cada estudiante solo puede marcar una única opción correcta por cada pregunta en cada examen
  - f) cada estudiante debe marcar al menos una respuesta para cada pregunta en un examen concreto
  - g) en un examen cualquiera puede haber preguntas duplicadas si tienen distinta puntuación

\* responder = marcar una o varias opciones de la pregunta como correctas

### 5) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

- a) DNI y nombre de todos los estudiantes que se han examinado alguna vez.

**RESPUESTAS x ESTUDIANTE donde es=dni [dni,nombre]**

- b) Enunciado de la pregunta (o preguntas) que ha tenido la puntuación más alta de cualquier examen.

**define alias C1 para contenido**

**define alias C2 para contenido**

**C1 [codpr]**

-

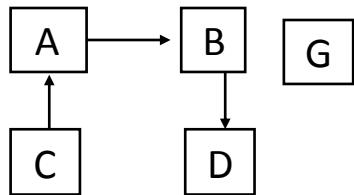
**( C1 x C2 donde C1.puntua < C2.puntua [C1.codpr] )**

**∞**

**PREGUNTA [enunciado]**

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula: \_\_\_\_\_ Tiempo TOTAL: 2 horas 45 min Puntuación: 10 (x4 = 40 puntos)

**1) Normaliza hasta FNBC (recuerda: tablas completamente definidas excepto dominios) (0/2)**



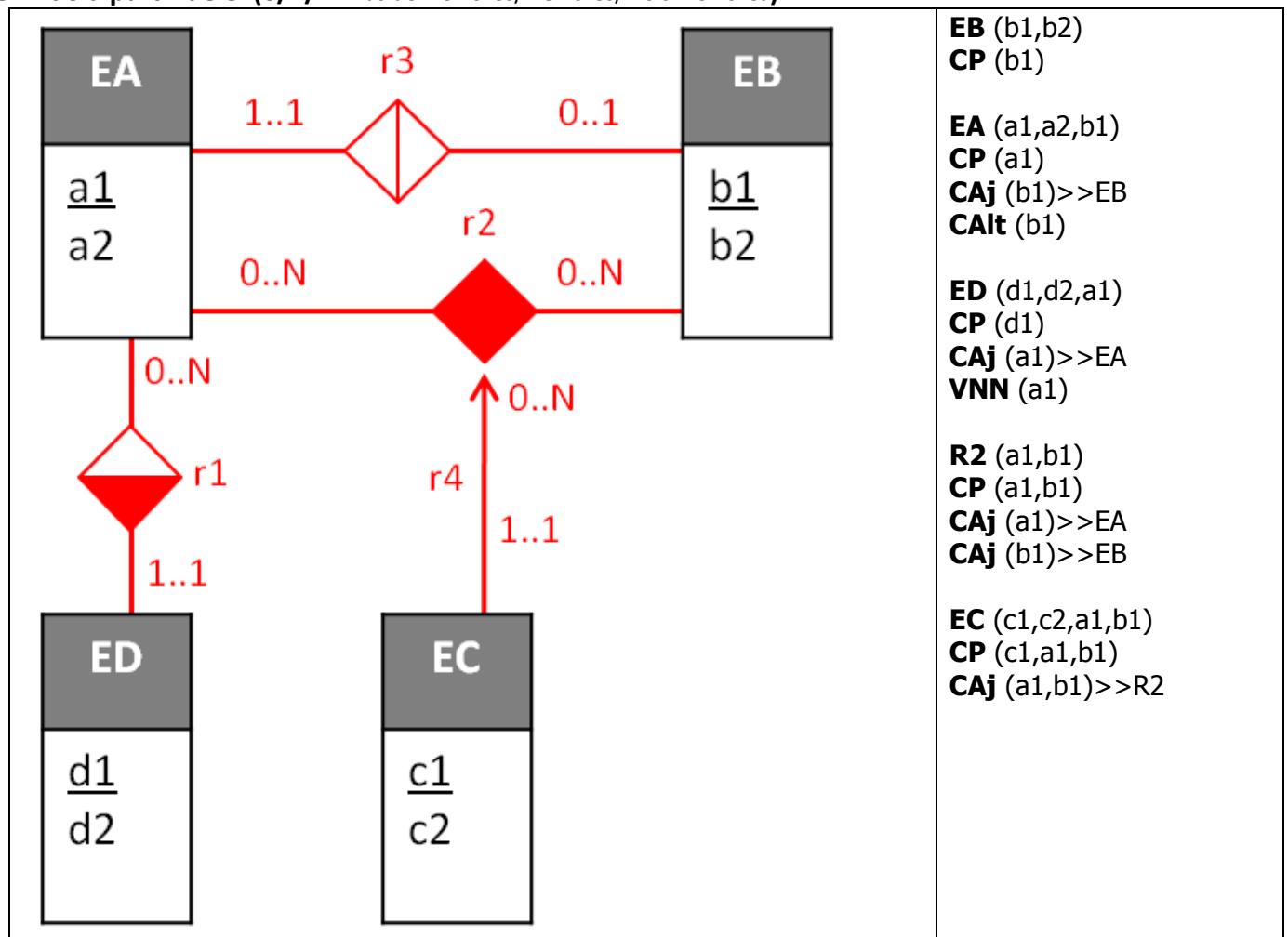
T(C,G) CP(C,G)  
 CAj(C)>>T1

T1(C,A) CP(C)  
 CAj(A)>>T2

T2(A,B) CP(A)  
 CAj(B)>>T3

T3(B,D) CP(B)

**2) Completa este esquema entidad-relación en un esquema relacional de acuerdo a las tablas que se han definido a partir de él (0/1/2=más de 2 errores/2 errores/1 o sin errores).**



3) Modifica o añade todas las restricciones necesarias para que el esquema relacional de base de datos siguiente sea conforme a las expresiones de correspondencia entre clases que se muestran a continuación (0/1/2=más de 2 errores/2 errores/1 o sin errores).

EVENTO (cod, descripción) CP (cod)	
DISTRITO (num) CP (num)	
CUADRANTE (cuad)	<b>CUADRANTE</b> (cuad, num) <b>CP</b> (num) <b>CAj</b> (num) >> DISTRITO por h
UNIDAD (nuni, nombre) CP (nuni) CAlt (nombre)	<b>UNIDAD</b> (nuni, nombre, num) <b>CP</b> (nuni) <b>CAlt</b> (nombre) <b>CAj</b> (num) >> CUADRANTE <b>VNN</b> (num) por d
LOCALIZACIÓN (num, extensión, tipo) CP(num)	<b>LOCALIZACIÓN</b> (num, extensión, tipo, unidad) <b>CP</b> (num) <b>CAj</b> (unidad) >> UNIDAD por e
SALIDA (línea, orientación) CP (línea)	<b>SALIDA</b> (línea, orientación, sal, cod) <b>CP</b> (sal, línea) <b>CAj</b> (sal) >> LOCALIZACIÓN por a) <b>CAj</b> (cod) >> EVENTO por b)
CATEGORÍA (concepto) CP (concepto)	<b>CATEGORÍA</b> (concepto, relacionado) <b>CAj</b> (relacionado) >> CATEGORÍA por c)
	<b>ACTUACIÓN</b> (nuni, cat) <b>CP</b> (nuni) <b>CAlt</b> (cat) <b>CAj</b> (nuni) → UNIDAD <b>CAj</b> (cat) → CATEGORÍA por g)
	<b>ESPECIALIZADA</b> (nuni, cat) <b>CP</b> (nuni, cat) <b>CAj</b> (nuni) → UNIDAD <b>CAj</b> (cat) → CATEGORÍA por f)

- a) Card(LOCALIZACIÓN, tiene)= (0,N)
- b) Card(EVENTO, por)= (0,N)
- c) Card(CATEGORÍA1, padre)= (0,N)
- d) Card(UNIDAD, vigila)= (1,1)
- e) Card(UNIDAD, sita)= (0,N)
- f) Card(UNIDAD, se\_especializa)= (0,N)
- g) Card(UNIDAD, actuación)= (0,1)
- h) Card(CUADRANTE, esta\_en)=(1,1)

- Card(SALIDA, tiene)= (1,1) DEPENDENCIA DE IDENTIFICADOR
- Card(SALIDA, por)= (0,1)
- Card(CATEGORÍA2, padre)= (0,1)
- Card(CUADRANTE, vigila)= (0,N)
- Card(LOCALIZACIÓN, sita)= (0,1)
- Card(CATEGORÍA, se\_especializa)= (0,N)
- Card(CATEGORÍA, actuación)= (0,1)
- Card(DISTRITO, esta\_en)= (0,1) GENERALIZACIÓN

**4) Da el enunciado de las consultas que se están resolviendo con estas expresiones en álgebra relacional  
(0/1/2=ninguna/1 correcta/2 correctas)**

a) EXAMINADO x ESTUDIANTE donde EXAMINADO.es=ESTUDIANTE.dni [dni,nombre]

DNI y nombre de los estudiantes que se han examinado alguna vez.

b)

define alias E1 para EXAMEN

define alias E2 para EXAMEN

E1 x E2 donde E1.pregunta = E2.pregunta y E1.examen <> E2.examen [E1.pregunta]

∞

PREGUNTA [enunciado]

Enunciado de las preguntas que han sido utilizadas en 2 exámenes o más.

**5) Dado el siguiente esquema y estado de base de datos (0/1/2=más de 2 errores/2 errores/1 o sin errores, entendiendo por error una orden cuyo resultado no es correcto)**

CLIENTE(dni,nombre) CP(dni)

EMPLEADO(ID,nom,atiende) CP(ID)

CAj(atiende)>>CLIENTE en borrados PROPAGAR modificaciones PROPAGAR

ORDEN(num,emp,cobro) CP(num,emp)

CAj(emp)>>EMPLEADO en borrados ANULAR modificaciones RECHAZAR

CLIENTE	
dni	nombre
21	Pepe
42	Ana
53	Luisa

EMPLEADO		
ID	nom	atiende
10	Elisa	21
11	Manuel	21
12	Jacobo	42

ORDEN		
num	emp	cobro
1	10	525
2	10	112
1	11	112

Después de ejecutar todas las órdenes siguientes:

1. delete from CLIENTE where dni=21;
2. update CLIENTE set dni=62 where dni=42;
3. delete from EMPLEADO where ID=11;
4. update ORDEN set emp=200 where emp=10;

¿cuál sería el estado final de la BD?

CLIENTE	
dni	nombre
21	Pepe
62	Ana
53	Luisa

EMPLEADO		
ID	nom	atiende
10	Elisa	21
11	Manuel	21
12	Jacobo	62

ORDEN		
num	emp	cobro
1	10	525
2	10	112
1	11	112

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 1 hora 15 min Puntuación: 8 (x5 = 40 puntos)

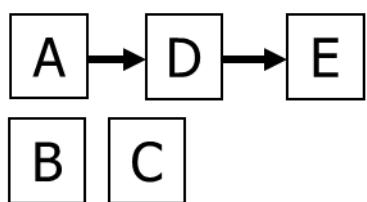
**1) Normaliza (0/1/2=más de 1 error / 1 error / sin errores)**

a) hasta FNBC (recuerda: tablas completamente definidas excepto dominios)

b) hasta 4FN; además de las dependencias funcionales expuestas, se dan las siguientes dependencias multivalor:

A >> B

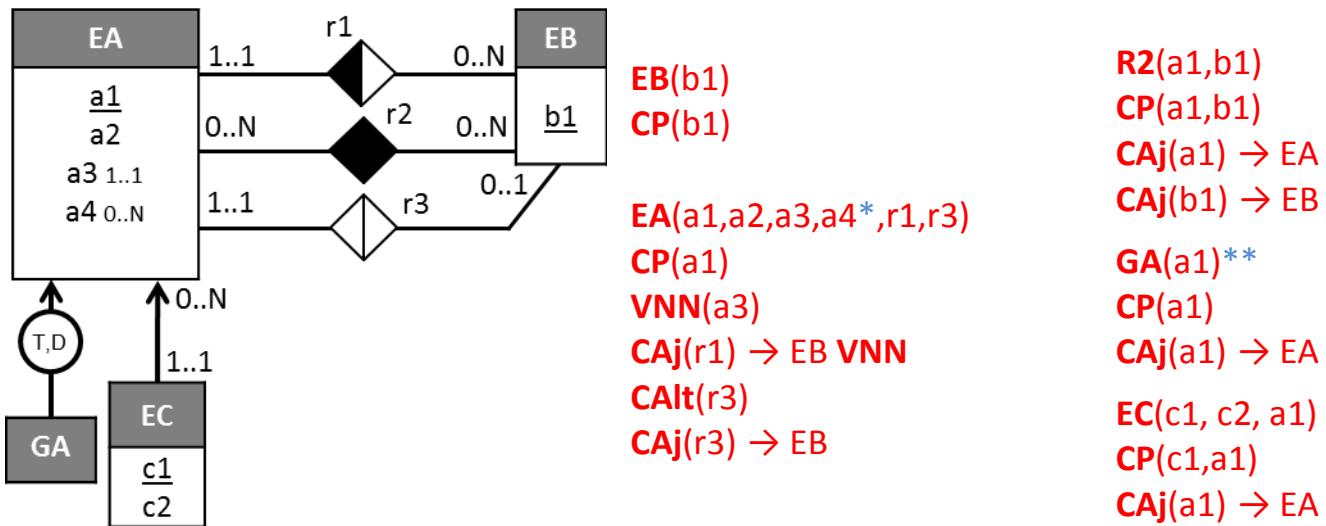
C >> B



- |                                                                  |                                                 |                      |
|------------------------------------------------------------------|-------------------------------------------------|----------------------|
| <b>a)</b><br>$T(a,b,c)$ CP( $a,b,c$ )<br>$CAj(a) \rightarrow T1$ | $T1(a,d)$<br>$CP(a)$<br>$CAj(d) \rightarrow T2$ | $T2(d,e)$<br>$CP(d)$ |
| <b>b)</b><br>$T(a,b)$ CP( $a,b$ )<br>$CAj(a) \rightarrow T1$     | $T3(c,b)$<br>$CP(c,b)$                          |                      |

La tabla T no está en 4FN y por ello se descompone, en el apartado b), en dos tablas. La clave ajena que ya tenía T por 3FN solo es coherente en esa tabla, no en T3.

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores)** No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



\* se pierde el atributo multivaluado (aunque se podría haber solucionado con otra tabla)

\*\* se pierden las propiedades de cobertura de la generalización (y se convierte en generalización parcial y solapada)

### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de grupos de desarrollo, formados por personas, y de los proyectos que dirigen y las reuniones que mantienen.

<b>MIEMBRO</b> (idM, nombre, grupo)	<b>GRUPO</b> (idG, denominación, líder)	<b>REUNIÓN</b> (fecha, grupo, lugar, proyecto)
<b>CP</b> (idM)	<b>CP</b> (idG)	<b>CP</b> (fecha, grupo)
<b>CAj</b> (grupo)→ GRUPO	<b>CAlt</b> (líder)	<b>CAj</b> (grupo)→ GRUPO
<b>LÍDER</b> (idM)	<b>CAj</b> (líder)→ LIDER	<b>CAj</b> (proyecto)→ PROYECTO
<b>CP</b> (idM)		<b>VNN</b> (proyecto)
<b>CAj</b> (idM)→ MIEMBRO		
<b>PROYECTO</b> (idP, título, lidera)		<b>ASISTENCIA</b> (miembro, reunF, reunG )
<b>CP</b> (idP)		<b>CP</b> (miembro, reunF, reunG )
<b>CAj</b> (lidera)→ GRUPO		<b>CAj</b> (miembro)→ MIEMBRO
		<b>CAj</b> (reunF, reunG)→ REUNIÓN

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                        |                                      |
|----------------------------------------|--------------------------------------|
| a) Card(MIEMBRO, pertenece) = ( 0, 1 ) | b) Card(GRUPO, pertenece) = ( 0, N ) |
| c) Card(LIDER, es) = ( 1, 1 )          | d) Card(MIEMBRO, es) = ( 0, 1 )      |
| e) Card(GRUPO, lidera) = ( 1, 1 )      | f) Card(LIDER, lidera) = ( 0, 1 )    |
| g) Card(PROYECTO, dirigido) = ( 0, 1 ) | h) Card(GRUPO, dirigido) = ( 0, N )  |
| i) Card(GRUPO, hace) = ( 0, N )        | l) Card(REUNIÓN, hace) = ( 1, 1 )    |
| j) Card(PROYECTO, sobre) = ( 0, N )    | m) Card(REUNIÓN, sobre) = ( 1, 1 )   |
| k) Card(MIEMBRO, asiste) = ( 0, N )    | n) Card(REUNIÓN, asiste) = ( 0, N )  |

### 4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

- a) Nombre de los miembros que no son líderes e identificador del grupo al que pertenece cada uno.

**MIEMBRO ×LIDER [nombre, grupo]**

Solución alternativa:

**MIEMBRO [idM] – (LIDER [idM]) × MIEMBRO [nombre, grupo]**

- b) Nombre de los miembros y fecha de las reuniones a las que asiste.

**MIEMBRO × (ASISTENCIA REN[miembro|idM]) [nombre, reunF]**

Solución alternativa:

**MIEMBRO ×<sub>( MIEMBRO.idM = ASISTENCIA.miembro )</sub> ASISTENCIA [nombre, reunF]**

Solución alternativa:

**MIEMBRO x ASISTENCIA**

**donde MIEMBRO.idM = ASISTENCIA.miembro**

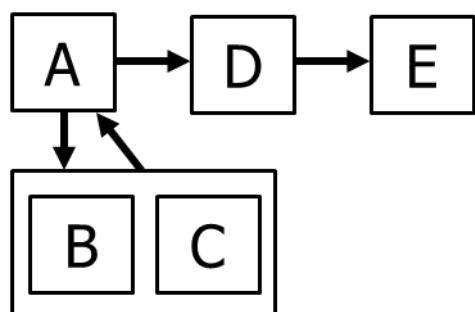
**[nombre, reunF]**

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo estimado: 1 h 15 min Puntuación: 8 (x5 = 40 puntos)

**1) Normaliza (0/1/2=más de 1 error / 1 error / sin errores)**

a) hasta FNBC (recuerda: tablas completamente definidas excepto dominios)

b) hasta 4FN; además de las dependencias funcionales expuestas, se da la siguiente dependencia multivalor:  $B >> C$

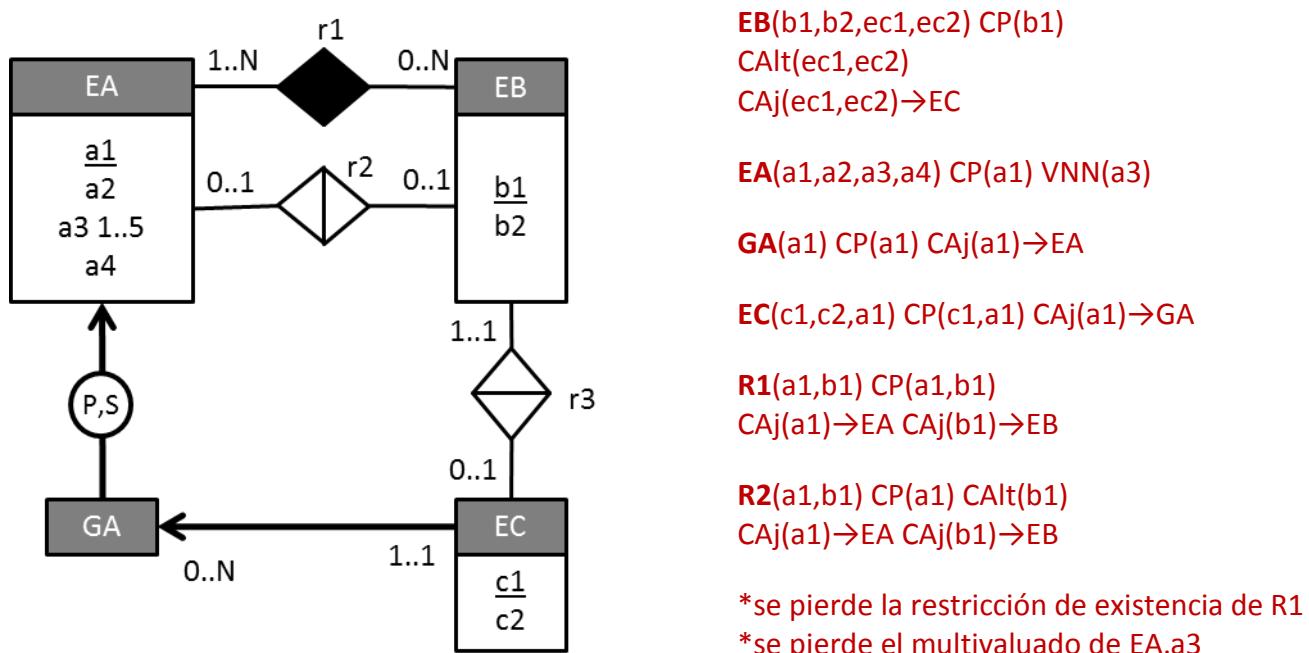


a)

T1(A,B,C,D)      T2 (D,E)  
 CP(A)                CP(D)  
 CA<sub>Alt</sub>(B,C)  
 CA<sub>j</sub>(D) → T2

b) No se necesita hacer nada, no hay problema de 4FN.

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores)** No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de grupos de desarrollo, formados por personas, y de los proyectos que dirigen y las reuniones que mantienen.

<b>MIEMBRO</b> (idM, nombre, grupo, lider)	<b>GRUPO</b> (idG, denominación, lider)	<b>REUNIÓN</b> (fecha, grupo, líder, lugar, proyecto)
<b>CP</b> (idM)	<b>CP</b> (idG, lider)	<b>CP</b> (fecha, grupo, lider)
<b>CAj</b> (grupo, lider)→ GRUPO	<b>CAj</b> (líder)→ LIDER	<b>CAj</b> (grupo, lider)→ GRUPO
<b>LÍDER</b> (idM)	<b>PROYECTO</b> (idP, idG, titulo, lider)	<b>CAj</b> (proyecto)→ PROYECTO
<b>CP</b> (idM)	<b>CP</b> (idP)	<b>ASISTENCIA</b> (miembro, reunF, reunG, reunL)
<b>CAj</b> (idM)→ MIEMBRO	<b>CAlt</b> (idG)	<b>CP</b> (miembro)
	<b>CAj</b> (idG, lider)→ GRUPO	<b>CAlt</b> (reunF, reunG, reunL)
		<b>CAj</b> (miembro)→ MIEMBRO
		<b>CAj</b> (reunF, reunG, reunL)→ REUNIÓN

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                     |                                   |
|-------------------------------------|-----------------------------------|
| a) Card(MIEMBRO, pertenece) = (0,1) | b) Card(GRUPO, pertenece) = (0,N) |
| c) Card(LIDER, es) = (1,1)          | d) Card(MIEMBRO, es) = (0,1)      |
| e) Card(GRUPO, lidera) = (1,1)      | f) Card(LIDER, lidera) = (0,N)    |
| g) Card(PROYECTO, dirigido) = (1,1) | h) Card(GRUPO, dirigido) = (0,1)  |
| i) Card(GRUPO, hace) = (0,N)        | l) Card(REUNIÓN, hace) = (1,1)    |
| j) Card(PROYECTO, sobre) = (0,N)    | m) Card(REUNIÓN, sobre) = (0,1)   |
| k) Card(MIEMBRO, asiste) = (0,1)    | n) Card(REUNIÓN, asiste) = (0,1)  |

4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, proporciona el enunciado correspondiente a las siguientes expresiones de álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

a) **MIEMBRO[idM] – (ASISTENCIA[miembro]) × MIEMBRO[nombre]**

Nombre de los miembros que nunca han asistido a una reunión

b) define alias R1 para REUNIÓN  
define alias R2 para REUNIÓN

REUNION donde proyecto = 'P1' [fecha] –  
(  
R1 × R2  
donde R1.proyecto = 'P1' and R2.proyecto = 'P1' and R1.fecha < R2.fecha  
[R1.fecha]  
)

Fecha de la última reunión del proyecto con identificador "P1".

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 1 h 15 min Puntuación: 8 (x5 = 40 puntos)

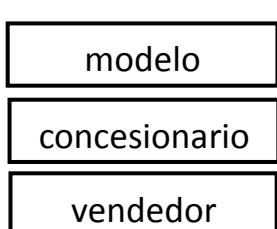
**1) Normaliza (0/1/2=más de 1 error / 1 error / sin errores)**

a) hasta FNBC (recuerda: tablas completamente definidas excepto dominios)

b) hasta 5FN; además de las dependencias funcionales expuestas, se dan las siguientes dependencias multivalor:

concesionario >> modelo

vendedor >> modelo

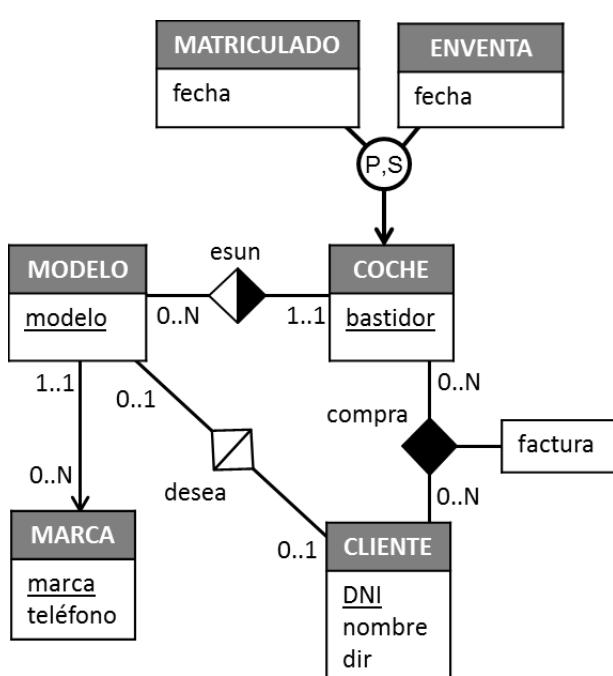


a) T(modelo, concesionario, vendedor)  
 CP(modelo, concesionario, vendedor)

b) T(modelo, concesionario) CP(modelo, concesionario)  
 T2(modelo, vendedor) CP(modelo, vendedor)

Se ha normalizado la tabla T a 4FN; no hay información adicional que obligue a normalizar hasta 5FN.

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores)** No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



**MATRICULADO**(bastidor, fecha)  
**ENVENTA**(bastidor, fecha)  
**CP**(bastidor)

**MODELO**(marca, modelo)  
**CP**(marca, modelo)  
**CAj**(marca) → **MARCA**

**COCHE**(bastidor, marca, modelo)  
**CP**(bastidor)  
**CAj**(marca, modelo) → **MODELO**  
**VNN**(marca)  
**VNN**(modelo)

**MATRICULADO**(bastidor, fecha)  
**CP**(bastidor)  
**CAj**(bastidor) → **COCHE**

**ENVENTA**(bastidor, fecha)  
**CP**(bastidor)  
**CAj**(bastidor) → **COCHE**

**CLIENTE**(DNI, nombre, dir)  
**CP**(DNI)

**COMPRA**(cliente, coche, factura)  
**CP**(cliente, coche)  
**CAj**(cliente) → **CLIENTE**  
**CAj**(bastidor) → **COCHE**

**DESEA**(cliente, marca, modelo)  
**CP**(cliente)  
**CAlt**(marca, modelo)  
**CAj**(cliente) → **CLIENTE**  
**CAj**(marca, modelo) → **MODELO**

**3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)**

Se trata de una base de datos de una empresa de compra-venta de coches usados y nuevos. Mantiene datos de qué coches tienen disponibles para la venta y de cuáles han sido vendidos y a quién.

<b>CLIENTE</b> (DNI, nombre, dirección)	<b>COCHE</b> (bastidor, marca, modelo)	<b>ENVENTA</b> (bastidor, fecha)
<b>CP</b> (DNI)	<b>CP</b> (bastidor)	<b>CP</b> (bastidor)
<b>MARCA</b> (marca, telef)	<b>CAj</b> (marca, modelo) → <b>MODELO</b>	<b>CAj</b> (bastidor) → <b>COCHE</b>
<b>CP</b> (marca)	<b>VNN</b> (marca)	<b>FACTURA</b> (numF, fecha, cliente, coche)
<b>MODELO</b> (marca, modelo)	<b>MATRICULADO</b> (bastidor, matricula, fecha)	<b>CP</b> (numF)
<b>CP</b> (marca, modelo)	<b>CP</b> (bastidor)	<b>CAj</b> (cliente) → <b>CLIENTE</b>
<b>CAj</b> (marca) → <b>MARCA</b>	<b>CAj</b> (bastidor) → <b>COCHE</b>	<b>CAj</b> (coche) → <b>MATRICULADO</b>
		<b>VNN</b> (cliente)

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                                 |                                                |
|-------------------------------------------------|------------------------------------------------|
| a) Card(COCHE, esun) = ( <b>1, 1</b> )          | b) Card(MODELO, esun) = ( <b>0, N</b> )        |
| c) Card(MODELO, de) = ( <b>1, 1</b> )           | d) Card(MARCA, de) = ( <b>0, N</b> )           |
| e) Card(COCHE, está) = ( <b>0, 1</b> )          | f) Card(MATRICULADO, está) = ( <b>1, 1</b> )   |
| g) Card(COCHE, disponible) = ( <b>0, 1</b> )    | h) Card(ENVENTA, disponible) = ( <b>1, 1</b> ) |
| i) Card(MATRICULADO, vendido) = ( <b>0, N</b> ) | k) Card(FACTURA, vendido) = ( <b>0, 1</b> )    |
| j) Card(CLIENTE, vendido_a) = ( <b>0, N</b> )   | l) Card(FACTURA, vendido_a) = ( <b>1, 1</b> )  |

**4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)**

- a) Número de bastidor de los coches que están para vender pero que no están matriculados.

**ENVENTA [bastidor] – (MATRICULADO [bastidor])**

- b) DNI y nombre de cliente, y marca y modelo de cualquier coche que se le haya facturado.

**CLIENTE x FACTURA x COCHE**  
**donde CLIENTE.DNI = FACTURA.cliente**  
**y FACTURA.coche = COCHE.bastidor**  
**[DNI, nombre, marca, modelo]**

Solución alternativa:

**CLIENTE ⋈<sub>(CLIENTE.DNI = FACTURA.cliente)</sub> FACTURA**  
**⋈<sub>(FACTURA.coche = COCHE.bastidor)</sub> COCHE [DNI, nombre, marca, modelo]**

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 1 h 15 min Puntuación: 8 (x5 = 40 puntos)

**1) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)**

Se trata de una base de datos de una empresa de compra-venta de coches usados y nuevos. Mantiene datos de qué coches tienen disponibles para la venta y de cuáles han sido vendidos y a quién.

<b>CLIENTE</b> (DNI, IDcliente, nombre)	<b>COCHE</b> (bastidor, modelo)	<b>MATRICULA</b> (matricula, fecha)
<b>CP</b> (DNI)	<b>CP</b> (bastidor)	<b>CP</b> (matricula)
<b>CAlt</b> (IDcliente)	<b>CAj</b> (modelo) → <b>MODELO</b>	<b>EXPOSICION</b> (bastidor, precio)
	<b>VNN</b> (modelo)	<b>CP</b> (bastidor)
<b>MARCA</b> (marca)	<b>PERMISO</b> (bastidor, matricula, cliente, fecha)	<b>CAj</b> : bastidor → <b>COCHE</b>
<b>CP</b> (marca)	<b>CP</b> (bastidor, matricula)	<b>FACTURA</b> (numF, fecha, cliente, coche, matricula, importe)
<b>MODELO</b> (marca, modelo)	<b>CAj</b> (bastidor) → <b>COCHE</b>	<b>CP</b> (numF)
<b>CP</b> (modelo)	<b>CAj</b> (matricula) → <b>MATRICULA</b>	<b>CAj</b> (cliente) → <b>CLIENTE</b>
<b>CAlt</b> (marca)	<b>CAj</b> (cliente) → <b>CLIENTE</b>	<b>CAj</b> (coche, matricula) → <b>PERMISO</b>
<b>CAj</b> (marca) → <b>MARCA</b>	<b>VNN</b> (cliente)	

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                            |                                              |
|--------------------------------------------|----------------------------------------------|
| a) Card(COCHE, esun) = <b>(1,1)</b>        | b) Card(MODELO, esun) = <b>(0,N)</b>         |
| c) Card(MODELO, de) = <b>(1,1)</b>         | d) Card(MARCA, de) = <b>(0,1)</b>            |
| e) Card(COCHE, permiso) = <b>(0,N)</b>     | f) Card(MATRICULA, permiso) = <b>(0,N)</b>   |
| g) Card(PERMISO, asignado) = <b>(1,1)</b>  | h) Card(CLIENTE, asignado) = <b>(0,N)</b>    |
| i) Card(COCHE, en_venta) = <b>(0,1)</b>    | k) Card(EXPOSICION, en_venta) = <b>(1,1)</b> |
| j) Card(CLIENTE, vendido_a) = <b>(0,N)</b> | l) Card(FACTURA, vendido_a) = <b>(0,1)</b>   |

**2) Álgebra Relacional (AR). Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve (0/1/2=ninguna/1 correcta/2 correctas)**

a) Proporciona la consulta AR para el siguiente enunciado:

Matrícula de los coches que han sido facturados a un cliente pero aún no tienen permiso asignado a ese cliente.

**FACTURA[coche,matricula,cliente]-(PERMISO[bastidor,matricula,cliente])[matricula]**

b) Proporciona el enunciado para la siguiente consulta AR:

Definir alias M1 para MATRICULA

Definir alias M2 para MATRICULA

MATRICULA[matricula]-(M1xM2 donde M1.fecha < M2.fecha [M1.matricula])

x PERMISO donde (MATRICULA.matricula=PERMISO.matricula)[bastidor]

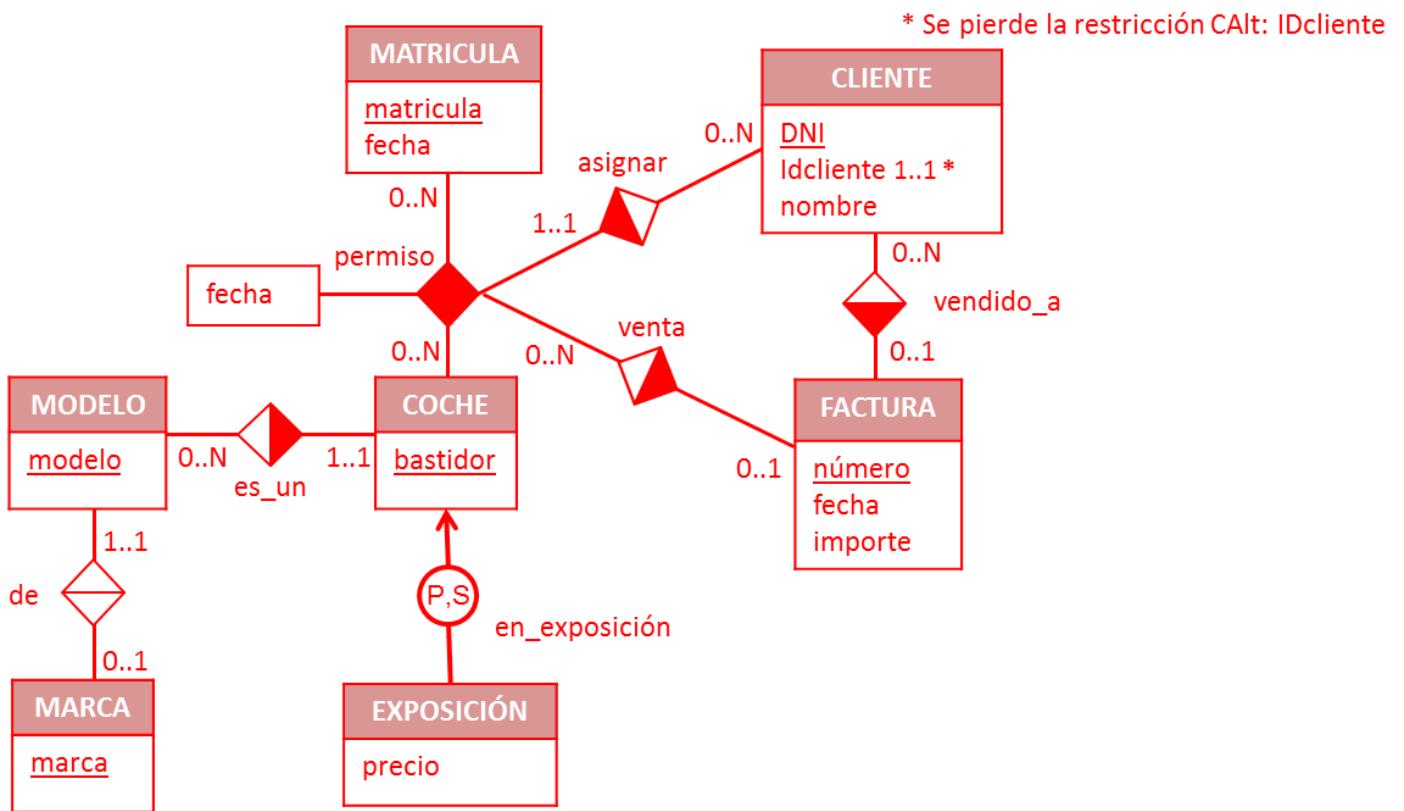
**Bastidor del último (o últimos) vehículo(s) matriculado(s)**

<b>CLIENTE</b> (DNI, IDcliente, nombre)	<b>COCHE</b> (bastidor, modelo)	<b>MATRICULA</b> (matricula, fecha)
<b>CP</b> (DNI)	<b>CP</b> (bastidor)	<b>CP</b> (matricula)
<b>CAlt</b> (IDcliente)	<b>CAj</b> (modelo) → <b>MODELO</b>	<b>EXPOSICION</b> (bastidor, precio)
<b>MARCA</b> (marca)	<b>VNN</b> (modelo)	<b>CP</b> (bastidor)
<b>CP</b> (marca)	<b>PERMISO</b> (bastidor, matricula, cliente, fecha)	<b>CAj</b> : bastidor → <b>COCHE</b>
<b>MODELO</b> (marca, modelo)	<b>CP</b> (bastidor, matricula)	<b>FACTURA</b> (numF, fecha, cliente, coche, matricula, importe)
<b>CP</b> (modelo)	<b>CAj</b> (bastidor) → <b>COCHE</b>	<b>CP</b> (numF)
<b>CAlt</b> (marca)	<b>CAj</b> (matricula) → <b>MATRICULA</b>	<b>CAj</b> (cliente) → <b>CLIENTE</b>
<b>CAj</b> (marca) → <b>MARCA</b>	<b>CAj</b> (cliente) → <b>CLIENTE</b>	<b>CAj</b> (coche, matricula) → <b>PERMISO</b>
	<b>VNN</b> (cliente)	

3) Responde con verdadero o falso (0/1/2=3 o más errores/2/hasta 1 error; no contestada=error)

- |          |                                                                                                                                           |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>V</b> | a) Un mismo coche puede tener varios permisos de circulación con diferentes matrículas                                                    |
| <b>F</b> | b) Podría repetirse el mismo modelo "R5" en la marca "RENAULT" y en la marca "SEAT"                                                       |
| <b>F</b> | c) Un vehículo facturado ya no puede estar en exposición                                                                                  |
| <b>F</b> | d) El cliente a quien se le asigna el permiso de circulación de un vehículo debe ser necesariamente el mismo a quien se le haya facturado |
| <b>F</b> | e) Un mismo coche podrá ser facturado a diferentes clientes sólo si cambia su matrícula                                                   |
| <b>V</b> | f) A un mismo cliente se le puede facturar varias veces el mismo coche en una misma fecha                                                 |
| <b>V</b> | g) Siempre puedo conocer la matrícula de cualquier coche facturado                                                                        |

4) Transforma el esquema relacional anterior en su equivalente esquema entidad-relación (0/1/2=más de 1 error / 1 error / sin errores) Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_

Aula y fecha: \_\_\_\_\_

Tiempo: 60 min

Puntuación: 8 (x5 = 40 puntos)

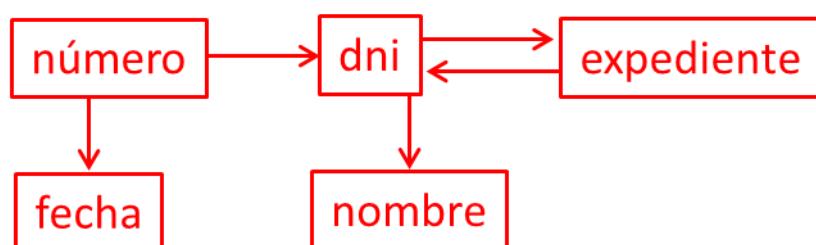
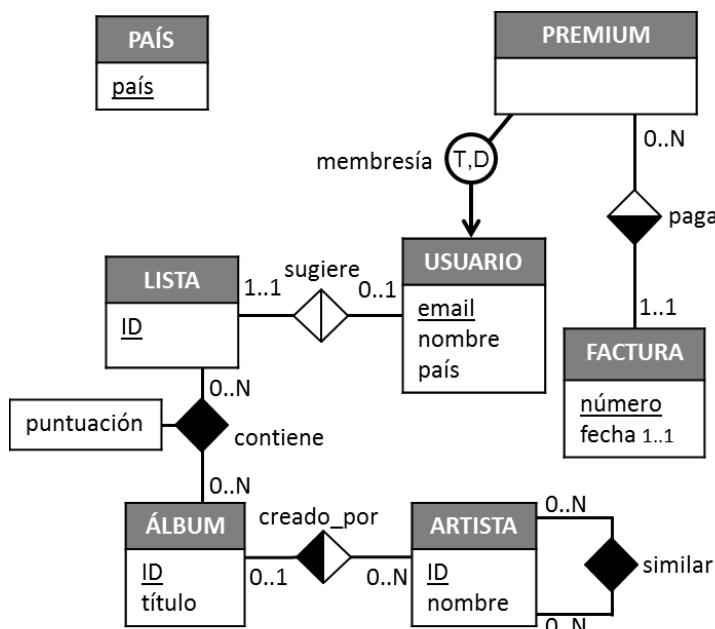
**1) Representa las dependencias funcionales que han dado lugar a estas tablas (0/2=más de 1 error / sin errores)**

De esta tabla, T(dni, expediente, nombre, número, fecha), tras aplicar todos los pasos necesarios para normalizar hasta FNBC se han obtenido estas otras tablas

**ALUMNO (dni, expediente, nombre)** CP (dni) CA<sub>Alt</sub>(expediente)

**MATRÍCULA (número, alumno, fecha)** CP(número) CA<sub>j</sub>(alumno) → ALUMNO

¿Qué dependencias funcionales había entre los atributos de T para que el esquema definitivo haya sido el de ALUMNO Y MATRÍCULA?

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 1 error / 1 error / sin errores)** No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.

PAÍS país	PAIS(país) CP(país)	LISTA(id, usuario) CP(id) CA <sub>Alt</sub> (usuario) CA <sub>j</sub> (usuario) → USUARIO
LISTA ID	USUARIO(email, nombre, país) CP(email)	ARTISTA(id, nombre) CP(id)
PREMIUM	PREMIUM(usuario) CP(usuario) CA <sub>j</sub> (usuario) → USUARIO	ALBUM(id, titulo, artista) CP(id) CA <sub>j</sub> (artista) → ARTISTA
USUARIO email nombre país	FACTURA(numero, fecha, usuario) CP(numero) VNN(fecha)	FACTURA(numero, fecha, usuario) CP(numero) VNN(usuario)
Membresía T,D	CA <sub>j</sub> (usuario) → PREMIUM	CONTIENE(lista, album, puntuacion) CP(lista, album) CA <sub>j</sub> (lista) → LISTA CA <sub>j</sub> (album) → ALBUM
sugiere 1..1, 0..1		SIMILAR(artist1, artist2) CP(artist1, artist2) CA <sub>j</sub> (artist1) → ARTISTA CA <sub>j</sub> (artist2) → ARTISTA
puntuación 0..N		
ÁLBUM ID título	creado_por 0..1, 0..N	
ARTISTA ID nombre	similar 0..N	

En la transformación a tablas se pierden las propiedades de cobertura de la generalización.

### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de una empresa de música en *streaming*. Mantiene datos de usuarios a los que se factura, listas personalizadas y álbumes que contienen canciones y los artistas autores de los anteriores.

<b>USUARIO</b> (email, nombre, pais) CP(email)	<b>LISTA</b> (id, usuario) CP(id, usuario) <b>CAj</b> (usuario) → USUARIO <b>CAj</b> (id) → ELEMENTO	<b>ARTISTA</b> (id, nombre) CP(id) <b>CAj</b> (id) → ELEMENTO
<b>FACTURA</b> (numero, numFiscal, fecha, email) CP(email, numero) <b>CAlt</b> (numFiscal) <b>CAj</b> (email) → USUARIO	<b>ELEMENTO</b> (id) CP(id)	<b>ALBUM</b> (id, titulo, artista) CP(id) <b>CAlt</b> (artista) <b>CAj</b> (artista) → ARTISTA <b>CAj</b> (id) → ELEMENTO
		<b>CANCION</b> (id, album, titulo) CP(id, album) <b>CAj</b> (album) → ALBUM <b>CAj</b> (id) → ELEMENTO

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                     |                                    |
|-------------------------------------|------------------------------------|
| a) Card(FACTURA, de) = ( 1, 1 )     | b) Card(USUARIO, de) = ( 0, N )    |
| c) Card(ELEMENTO, lista) = ( 0, N ) | d) Card(USUARIO, lista) = ( 0, N ) |
| e) Card(ARTISTA, es) = ( 1, 1 )     | f) Card(ELEMENTO, es) = ( 0, 1 )   |
| g) Card(ALBUM, de) = ( 1, 1 )       | h) Card(ARTISTA, de) = ( 0, 1 )    |
| i) Card(CANCION, en) = ( 1, 1 )     | j) Card(ALBUM, en) = ( 0, N )      |

### 4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

- a) select nombre from usuario where email in (select email from factura where fecha = '2016-06-09').

**FACTURA donde fecha='2016-06-09' × USUARIO [nombre]**

Solución alternativa:

**USUARIO × (FACTURA donde fecha='2016-06-09') [nombre]**

Solución alternativa:

**USUARIO x FACTURA donde (USUARIO.email=FACTURA.email y fecha='2016-06-09') [nombre]**

- b) select m.titulo, a.nombre from album m, artista a where m.artista = a.id.

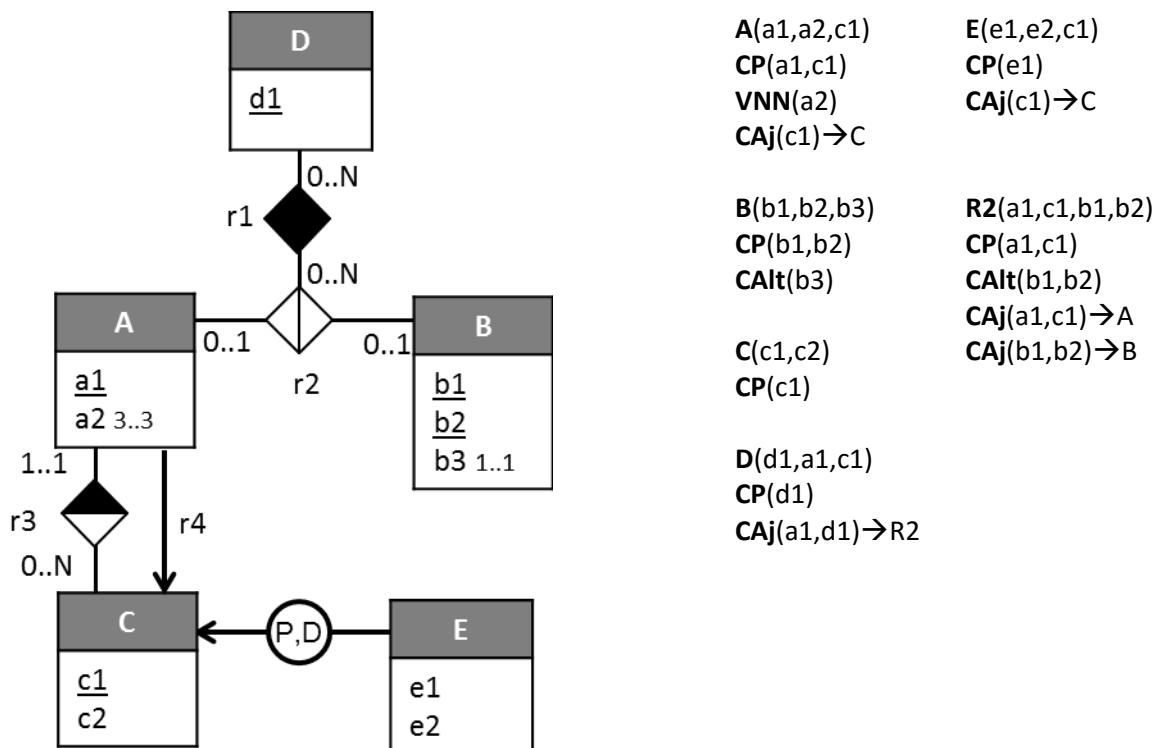
**ALBUM x ARTISTA**

**donde ALBUM.artista = ARTISTA.id**

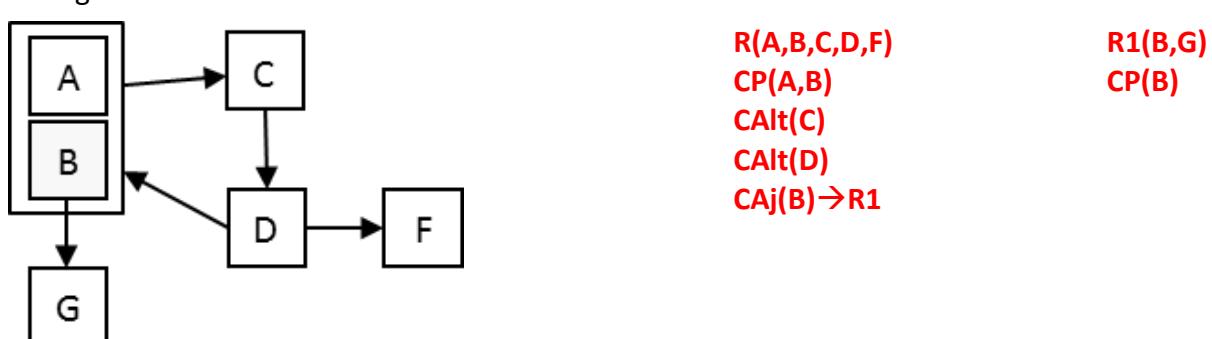
**[titulo, nombre]**

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 75 min Puntuación: 8 (x5 = 40 puntos)

- 1) Enumera los fallos o pérdidas cometidos al transformar el esquema entidad-relación en el esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No se tendrá en cuenta la definición de dominios para los atributos.



- 1) La restricción A.a2 3..3 no se puede reflejar en MR  
 2) La propiedad de cobertura D de la generalización no se puede reflejar en MR  
 3) Faltaría un atributo A.c11 CAj(c11) → C VNN(c11) para reflejar r3  
 4) En E debe ser CP(c1)  
 5) R1 está mal representado. En D sobran los atributos D.a1 y D.c1 así como la CAj(a1,d1) → R2. Faltaría una nueva tabla R1(a1,c1,d1) CP(a1,c1,d1) CAj(a1,c1) → R2 CAj(d1) → D  
 6) En B, no es CAlt(b3) sino VNN(b3)
- 2) Obtén el esquema relacional normalizado hasta la Forma Normal de Boyce Codd a partir del siguiente diagrama de dependencias funcionales (0/2)=con errores / sin errores) No es necesario definir dominios para ningún atributo.



### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de red social para relacionar mascotas y a sus propietarios con otras mascotas y propietarios compatibles.

**USUARIO**(email,nick, nombre, ciudad)  
**CP**(email)  
**CAlt**(nick)

**MASCOTA**(usuario, nombre, fechanac, sexo, raza)  
**CP**(usuario, nombre)  
**CAj**(usuario) → USUARIO

**FAVORITO**(usuario, mascota, dueño)  
**CP**(usuario, mascota, dueño)  
**CAj**(usuario) -> USUARIO  
**CAj**(dueño, mascota) -> MASCOTA

**CITA**(dueño1, mascota1, dueño2, mascota2, fecha)  
**CP**(dueño1, mascota1, fecha)  
**CAlt**(dueño2, mascota2, fecha)

**CAj**(dueño1, mascota1) → MASCOTA  
**CAj**(dueño2, mascota2) → MASCOTA

**ANUNCIO** (codigo, titulo, contenido, autor, fecha)  
**CP**(codigo)  
**VNN**(autor)  
**CAj**(autor) -> USUARIO

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                              |                                              |
|----------------------------------------------|----------------------------------------------|
| a) Card(MASCOTA, propietario) = <b>(1,1)</b> | b) Card(USUARIO, propietario) = <b>(0,N)</b> |
| c) Card(MASCOTA, favorito) = <b>(0,N)</b>    | d) Card(USUARIO, favorito) = <b>(0,N)</b>    |
| e) Card(MASCOTA1, participa1) = <b>(0,N)</b> | g) Card(CITA, participa1) = <b>(1,1)</b>     |
| f) Card(CITA, participa2)= <b>(1,1)</b>      | h) Card(MASCOTA2, participa2)= <b>(0,N)</b>  |
| i) Card(ANUNCIO, publica) = <b>(1,1)</b>     | j) Card(USUARIO, publica) = <b>(0,N)</b>     |

### 4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

- a) Nick de los usuarios que no tienen mascotas

**USUARIO [email] – (MASCOTA [usuario]) ⊗ USUARIO [nick]**

- b) Email de los usuarios que han conseguido una cita con una mascota marcada como favorita

**FAVORITO x CITA donde (usuario=dueño1 y dueño = dueño2 y mascota=mascota2) o (usuario=dueño2 y dueño = dueño1 y mascota=mascota1) [usuario]**

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 60 min Puntuación: 8 (x5 = 40 puntos)

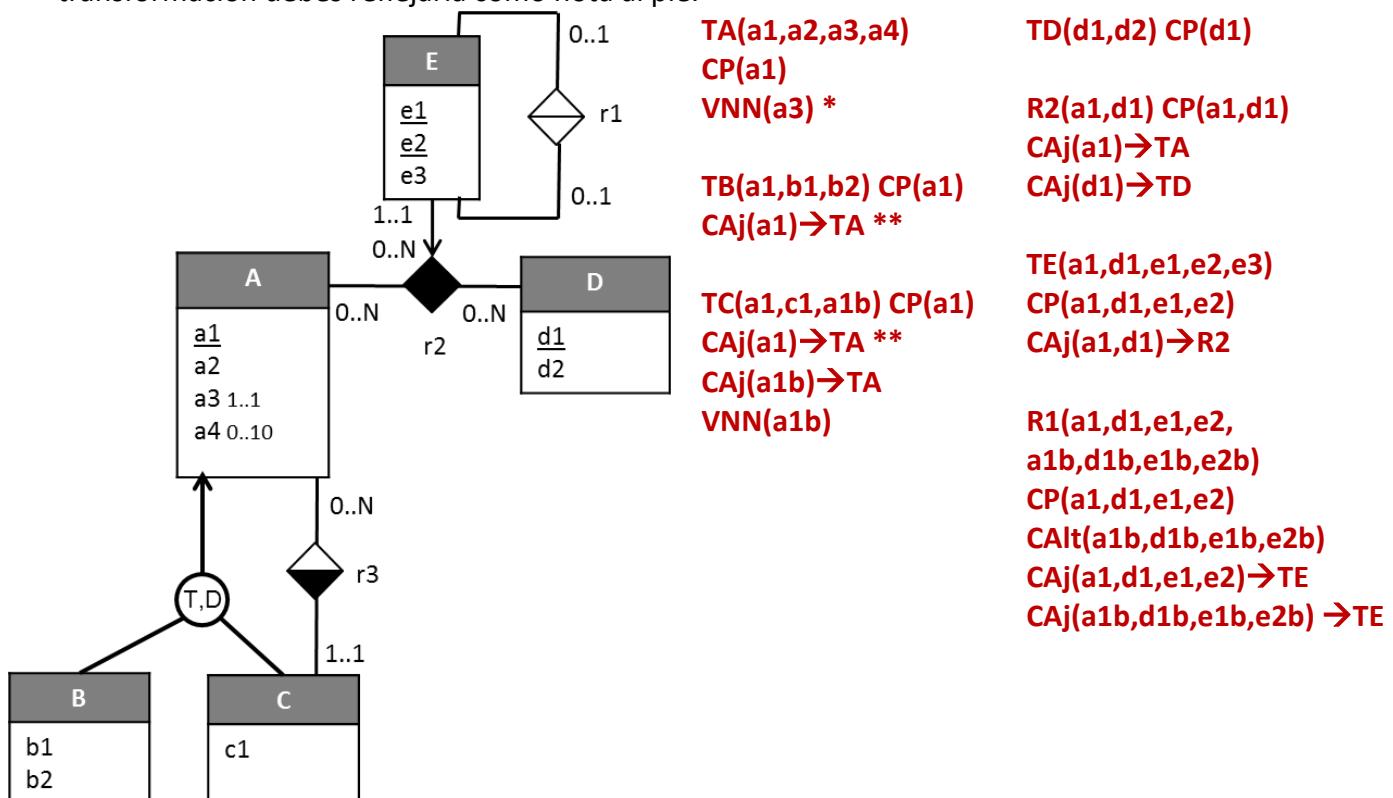
1) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

PACIENTE(dni,sip, nombre, dirección, teléfono) CP(dni) CAlt(sip)	MEDICO(dni,nombre, teléfono) CP(dni)	ESPECIALISTA(dni,especialidad) CP(dni) CAj(dni) → MEDICO CAj(especialidad) → ESPECIALIDAD VNN(especialidad)	SALA(planta, número, descripción) CP(planta,número)
HISTORIA(dni,orden, tratamiento) CP(dni,orden) CAj(dni) → PACIENTE	DEFAMILIA(dni) CP(dni) CAj(dni) → MEDICO	ESPECIALIDAD(idE, descripción) CP(idE) VNN(descripción)	CITA(paciente,médico, hora, fecha,salap,salan) CP(paciente,médico) CAj(paciente) → PACIENTE CAj(médico) → MEDICO CAj(salap,salan) → SALA VNN(hora) VNN(fecha)

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                             |                                          |
|---------------------------------------------|------------------------------------------|
| a) Card(PACIENTE, tiene) = <b>(0,N)</b>     | b) Card(HISTORIA, tiene) = <b>(1,1)</b>  |
| c) Card(MEDICO, es) = <b>(0,1)</b>          | d) Card(ESPECIALISTA, es) = <b>(1,1)</b> |
| e) Card(ESPECIALISTA, en) = <b>(1,1)</b>    | f) Card(ESPECIALIDAD, en) = <b>(0,N)</b> |
| g) Card(ESPECIALISTA, jefe ) = <b>(0,1)</b> | h) Card(SERVICIO, jefe) = <b>(1,1)</b>   |
| i) Card(PACIENTE,citado) = <b>(0,N)</b>     | j) Card(MEDICO, citado) = <b>(0,N)</b>   |
| k) Card(CITA,serEn) = <b>(0,1)</b>          | l) Card(SALA, serEn)= <b>(0,N)</b>       |

2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



\*Se pierde el multivaluado de TA.a4.

\*\* Se pierden las propiedades de cobertura T,D que pasan a ser P,S.

- 3) Dado el siguiente esquema y estado de base de datos (0/1/2=más de 2 errores/2 errores/1 o sin errores, entendiendo por error una orden cuyo resultado no es correcto)

**MARCA**(ID,nombre) **CP**(ID)

**PRODUCTO**(pcod,nom,marca) **CP**(pcod) **CAj**(marca)>>**MARCA** en borrados **PROPAGAR**  
modificaciones **PROPAGAR**

**PARTE**(num,prod,cobro) **CP**(num,prod) **CAj**(prod)>>**PRODUCTO** en borrados **ANULAR**  
modificaciones **RECHAZAR**

<b>MARCA</b>	
ID	nombre
11	Sony
22	Samsung
55	LG

<b>PRODUCTO</b>		
pcod	nom	marca
01	UBP-X800	11
02	UHP-H1	11
03	Bd-h6500	22

<b>PARTES</b>		
num	prod	cobro
1	01	50
2	01	30
1	02	30

Después de ejecutar todas las órdenes siguientes:

1. ~~delete from MARCA where ID=11;~~
2. update MARCA set ID=62 where ID=22;
3. ~~delete from PRODUCTO where pcod=02;~~
4. ~~update PARTE set prod=200 where prod=01;~~

¿cuál sería el estado final de la BD?

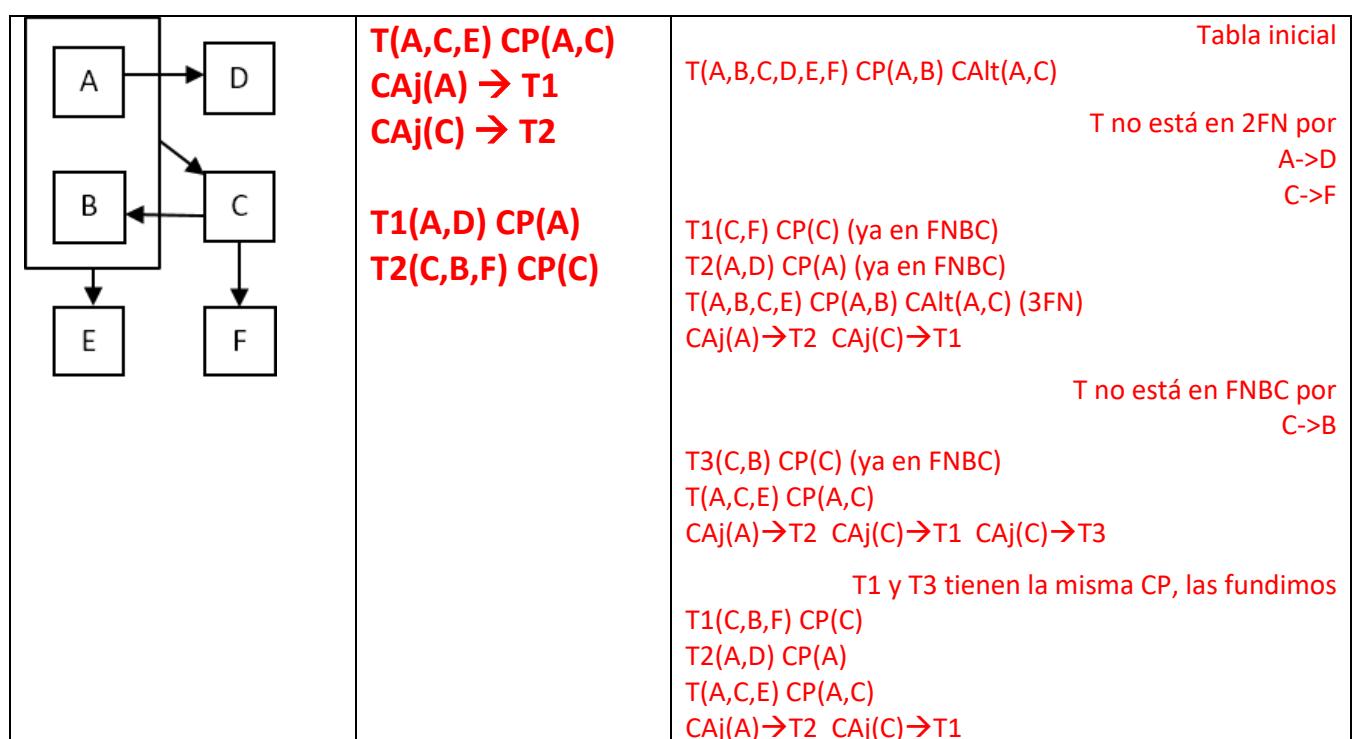
<b>MARCA</b>	
ID	nombre
11	Sony
62	Samsung
55	LG

<b>PRODUCTO</b>		
pcod	nom	marca
01	UBP-X800	11
02	UHP-H1	11
03	Bd-h6500	62

<b>PARTES</b>		
num	prod	cobro
1	01	50
2	01	30
1	02	30

- 4) Normaliza hasta FNBC (0/1/2=más de 1 error / 1 error / sin errores)

(recuerda: tablas completamente definidas excepto dominios)



D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo (de toda la recuperación): **2 h 45 min** Puntuación: **8** ( $\times 5 = 40$  puntos)

**1) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)**

<b>PRODUCTO</b> (idPro,precio, nombre, descripción, marca, vende)  <b>CP</b> (idPro) <b>CAj</b> (vende)→ <b>USUARIO</b> <b>VNN</b> (vende)  <b>USUARIO</b> (email,nombre, apellidos, contraseña, cpostal)  <b>CP</b> (email)	<b>INTERÉS</b> (idPro,usuario) <b>CP</b> (idPro, usuario) <b>CAj</b> (idPro)→ <b>PRODUCTO</b> <b>CAj</b> (usuario)→ <b>USUARIO</b>  <b>OPERACIÓN</b> (idPro, fecha, comprador) <b>CP</b> (idPro, fecha) <b>CAlt</b> (idPro, comprador) <b>CAj</b> (idPro) → <b>PRODUCTO</b> <b>CAj</b> (comprador) → <b>USUARIO</b>	<b>RECLAMACION</b> (idRec, idPro, fecha, fechaRec) <b>CP</b> (idRec) <b>CAlt</b> (idPro, fecha) <b>CAj</b> (idPro, fecha) → <b>OPERACIÓN</b>  <b>BUENO</b> (email) <b>CP</b> (email) <b>CAj</b> (email) → <b>USUARIO</b>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                                         |                                                       |
|---------------------------------------------------------|-------------------------------------------------------|
| a) Card( <b>PRODUCTO</b> , vende) = <b>(1,1)</b>        | b) Card( <b>USUARIO</b> , vende) = <b>(0,N)</b>       |
| c) Card( <b>PRODUCTO</b> , interés) = <b>(0,N)</b>      | d) Card( <b>USUARIO</b> , interés) = <b>(0,N)</b>     |
| e) Card( <b>OPERACIÓN</b> , comprar) = <b>(1,1)</b>     | f) Card( <b>USUARIO</b> , comprar) = <b>(0,N)</b>     |
| g) Card( <b>OPERACIÓN</b> , ofertar) = <b>(1,1)</b>     | h) Card( <b>PRODUCTO</b> , ofertar) = <b>(0,N)</b>    |
| i) Card( <b>RECLAMACIÓN</b> , denunciar) = <b>(1,1)</b> | j) Card( <b>OPERACIÓN</b> , denunciar) = <b>(0,1)</b> |
| k) Card( <b>USUARIO</b> ,es) = <b>(0,1)</b>             | l) Card( <b>BUENO</b> , es)= <b>(1,1)</b>             |

**2) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)**

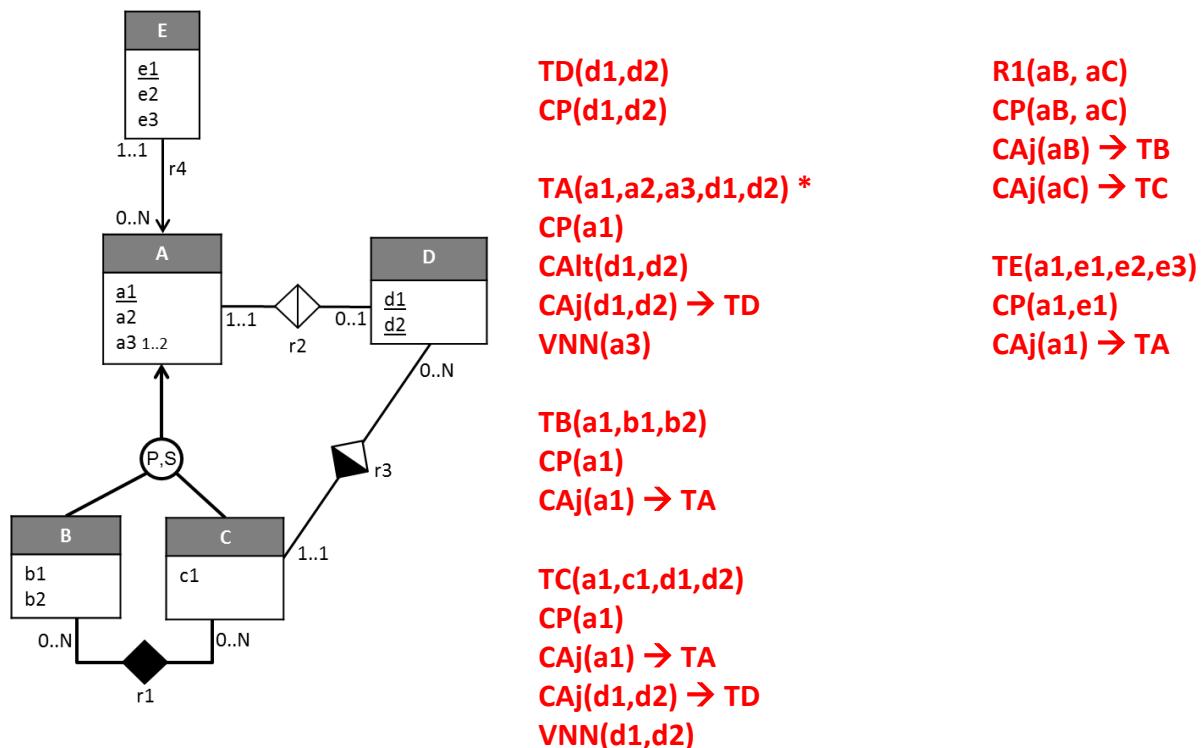
- a) select max(precio) from **PRODUCTO**

```
define alias P1 para PRODUCTO
define alias P2 para PRODUCTO
PRODUCTO[precio]
-
(P1 x P2 donde P1.precio < P2.precio [P1.precio])
```

- b) select u.email,u.nombre from **PRODUCTO** p join **USUARIO** u on (p.vende=u.email) where p.marca = 'CANON'

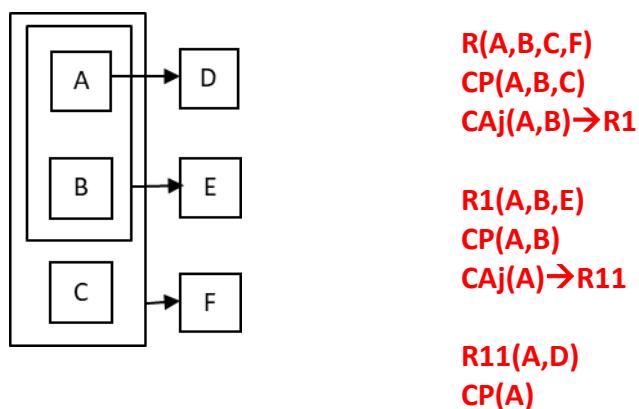
**PRODUCTO x USUARIO donde vende=email y marca = 'CANON' [email,USUARIO.nombre]**

- 3) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



\* se pierde multivaluado

- 4) Normaliza hasta FNBC (0/1/2=más de 1 error / 1 error / sin errores). Nota: tablas completamente definidas excepto dominios.



D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 60 min Puntuación: 8 (x5 = 40 puntos)

1) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

<b>MIEMBRO</b> (dni, licencia, nombre) <b>CP</b> (dni) <b>CAlt</b> (licencia)	<b>CLASE</b> (clase) <b>CP</b> (clase)	<b>MONITOR</b> (dni, titulacion) <b>CP</b> (dni) <b>CAj</b> (dni) >> MIEMBRO
<b>TRIKID</b> (dni, edad, clase, grupo) <b>CP</b> (dni) <b>CAj</b> (dni) >> MIEMBRO <b>CAj</b> (clase, grupo) >> GRUPO <b>VNN</b> (clase) <b>VNN</b> (grupo)	<b>GRUPO</b> (clase, grupo, monitor) <b>CP</b> (clase, grupo) <b>CAj</b> (clase) >> CLASE <b>CAj</b> (monitor) >> MONITOR <b>PRUEBA</b> (fecha, lugar, líder) <b>CP</b> (fecha, lugar) <b>CAj</b> (líder) >> MONITOR	<b>INSCRIPCION</b> (fecha, lugar, trikid) <b>CP</b> (fecha, lugar, trikid) <b>CAj</b> (fecha, lugar) >> PRUEBA <b>CAj</b> (trikid) >> TRIKID <b>PODIUM</b> (fecha, lugar, trikid, posicion) <b>CP</b> (fecha, lugar, posicion) <b>CAlt</b> (fecha, lugar, trikid) <b>CAj</b> (fecha, lugar, trikid) >> INSCRIPCION

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                           |                                          |
|-------------------------------------------|------------------------------------------|
| a) Card(TRIKID, es) = <b>(1,1)</b>        | b) Card(MIEMBRO, es) = <b>(0,1)</b>      |
| c) Card(GRUPO, entrena) = <b>(0,1)</b>    | d) Card(MONITOR, entrena) = <b>(0,N)</b> |
| e) Card(CLASE, pertenece) = <b>(0,N)</b>  | f) Card(GRUPO, pertenece) = <b>(1,1)</b> |
| g) Card(TRIKID, inscrito) = <b>(0,N)</b>  | h) Card(PRUEBA, inscrito) = <b>(0,N)</b> |
| i) Card(INSCRIPCION, hace) = <b>(0,1)</b> | j) Card(PODIUM, hace) = <b>(1,1)</b>     |
| k) Card(MONITOR, lidera) = <b>(0,N)</b>   | l) Card(PRUEBA, lidera) = <b>(0,1)</b>   |

2) Dado el siguiente esquema y estado de base de datos (0/1/2=más de 2 errores/2 errores/1 o sin errores, entendiendo por error una orden cuyo resultado no es correcto)

**ESTUDIANTE**(ID,nombre) **CP**(ID)

**EXAMEN**(IDX,estudiante,fecha) **CP**(IDX,estudiante)

**CAj**(estudiante)>>**ESTUDIANTE** borrados PROPAGAR modificaciones PROPAGAR

**CALIFICAR**(idx,est,calificador,nota) **CP**(idx,est)

**CAj**(calificador)>>**ESTUDIANTE** borrados ANULAR modificaciones PROPAGAR

**CAj**(idx,est)>>**EXAMEN** borrados RECHAZAR modificaciones PROPAGAR

<b>ESTUDIANTE</b>	
ID	nombre
11	Ana
22	Susana
55	Maite

<b>EXAMEN</b>		
IDX	estudiante	fecha
01	11	Marzo
02	11	Marzo
01	55	Marzo

<b>CALIFICAR</b>			
Idx	est	calificador	nota
01	11	22	50
01	11 <sup>1</sup>	55	30
01	55	22	30

Después de ejecutar todas las órdenes siguientes:

1. delete from ESTUDIANTE where ID=11; **rechazada por CALIFICAR.(idx,est)**
2. delete from ESTUDIANTE where ID=22; **ok**
3. update EXAMEN set IDX='03',estudiante=11 where IDX='01' and estudiante=55; **ok**
4. delete from CALIFICAR where nota>30; **ok**

¿cuál sería el estado final de la BD? (marca nulos como "null" o "nulo"; las filas completamente en blanco se entienden inexistentes)

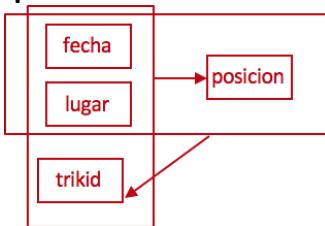
<b>ESTUDIANTE</b>	
ID	nombre
11	Ana
55	Maite

<b>EXAMEN</b>		
IDX	estudiante	fecha
01	11	Marzo
02	11	Marzo
03	11	Marzo

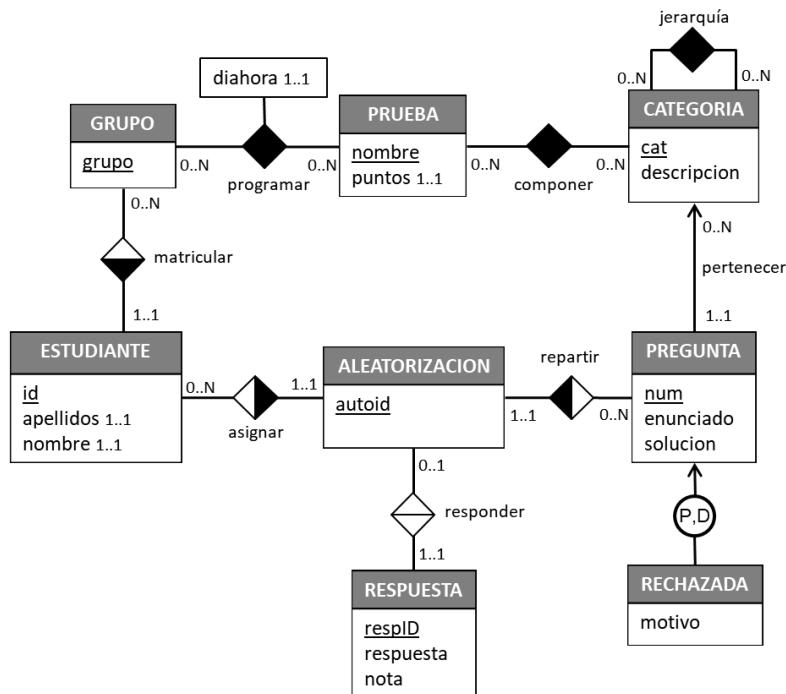
<b>CALIFICAR</b>			
idx	est	calificador	nota
01	11	55	30
03	11	null	30

<sup>1</sup> Hay un error en los datos mostrados en la tabla CALIFICAR, puesto que el valor ('01',11) contraviene las restricciones de clave primaria. Como no afecta al ejercicio, se ignora, y se responde y califica como si fuera correcto.

- 3) Si consideramos que la tabla PODIUM del esquema lógico del ejercicio 1 está correctamente normalizada en FNBC, ¿cuál sería su diagrama de dependencias funcionales? (0/2=más de 1 error / sin errores)



- 4) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



GRUPO(grupo) CP(grupo)

PRUEBA(nombre,puntos)  
CP(nombre)  
VNN(puntos)

PROGRAMAR(grupo, prueba,  
diahora)  
CP(grupo,prueba)  
CAj(grupo)>>GRUPO  
CAj(prueba)>>PRUEBA  
VNN(diahora)

CATEGORIA(cat,descripcion)  
CP(cat)

COMPONER(cat,prueba)  
CP(cat,prueba)  
CAj(cat)>>CATEGORIA  
CAj(prueba)>>PRUEBA

JERARQUIA(cat1,cat2)

CP(cat1,cat2)  
CAj(cat1)>>CATEGORIA  
CAj(cat1)>>CATEGORIA

ESTUDIANTE(id, apellidos, nombre,  
grupo)  
CP(id)  
CAj(grupo)>>GRUPO  
VNN(grupo)  
VNN(apellidos)  
VNN(nombre)

PREGUNTA(cat, num, enunciado,  
solucion)  
CP(cat,num)  
CAj(cat)>>CATEGORIA

RECHAZADA(cat, num, motivo)

CP(cat,num)  
CAj(cat,num)>>PREGUNTA  
\*No se mantiene la propiedad  
“disjunta” de la generalización

ALEATORIZA(autoID, cat, num, est)  
CP(autoID)  
CAj(cat, num)>>PREGUNTA  
CAj(est)>>ESTUDIANTE  
VNN(cat, num)  
VNN(est)

RESPUESTA(respID, respuesta, nota,  
aleat)  
CP(respID)  
CAlt(aleat)  
CAj(aleat)>>ALEATORIZA

D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 60min Puntuación: 8 (x5 = 40 puntos)

**1) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)**

<b>GRUPO(grupo)</b> <b>CP(grupo)</b>	<b>JERARQUIA(cat1,cat2)</b> <b>CP(cat1,cat2)</b> <b>CAj(cat1)&gt;&gt;CATEGORIA</b> <b>CAj(cat2)&gt;&gt;CATEGORIA</b>	<b>ALEATORIZA(autoid, cat, num, est)</b> <b>CP(autoid)</b> <b>CAj(cat, num)&gt;&gt;PREGUNTA</b> <b>CAj(est)&gt;&gt;ESTUDIANTE</b> <b>VNN(cat, num)</b> <b>VNN(est)</b>
<b>PRUEBA(nombre,puntos)</b> <b>CP(nombre)</b> <b>VNN(puntos)</b>	<b>ESTUDIANTE(id, apellidos, nombre, grupo)</b> <b>CP(id)</b> <b>CAj(grupo)&gt;&gt;GRUPO</b> <b>VNN(grupo)</b>	<b>RESPUESTA(respID, respuesta, nota, aleat)</b> <b>CP(respID)</b> <b>CAlt(aleat)</b> <b>CAj(aleat)&gt;&gt;ALEATORIZA</b>
<b>PROGRAMAR(grupo, prueba, diahora)</b> <b>CP(grupo,prueba)</b> <b>CAj(grupo)&gt;&gt;GRUPO</b> <b>CAj(prueba)&gt;&gt;PRUEBA</b> <b>VNN(diahora)</b>	<b>PREGUNTA(cat, num, enunciado, solucion)</b> <b>CP(cat,num)</b> <b>CAj(cat)&gt;&gt;CATEGORIA</b>	
<b>CATEGORIA(cat,descripcion)</b> <b>CP(cat)</b>		

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                        |                                         |
|----------------------------------------|-----------------------------------------|
| a) Card(GRUPO, matricular) = (0,N)     | b) Card(ESTUDIANTE, matricular) = (1,1) |
| c) Card(GRUPO, programar) = (0,N)      | d) Card(PRUEBA, programar) = (0,N)      |
| e) Card(ALEATORIZA, asignar) = (1,1)   | f) Card(ESTUDIANTE, asignar) = (0,N)    |
| g) Card(PREGUNTA, pertenece) = (1,1)   | h) Card(CATEGORÍA, pertenece) = (0,N)   |
| i) Card(ALEATORIZA, repartir) = (1,1)  | j) Card(PREGUNTA, repartir) = (0,N)     |
| k) Card(CATEGORIA1, jerarquía) = (0,N) | l) Card(CATEGORÍA2, jerarquía)= (0,N)   |

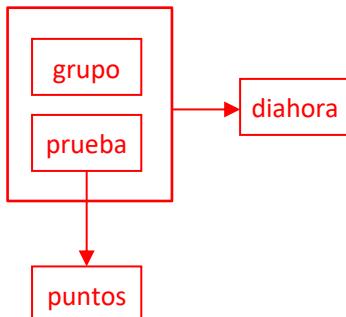
**2) Partiendo del esquema anterior.**

- a) ¿Cuál sería la tabla única de la que derivarían GRUPO, PRUEBA y PROGRAMAR si estuvieran en 1FN? b)  
 ¿Cuál sería su diagrama de dependencias funcionales?  
 (0/1/2=ninguna / 1 correcta / 2 correctas)

- a) PROGRAMAR (grupo, prueba, diahora, puntos)

CP (grupo, prueba)

b)



- 3) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

a) Nombre y apellidos de los estudiantes que les han asignado (aleatoriza) una pregunta sin solución (cadena vacía "")

ALEATORIZA  $\bowtie$  PREGUNTA  $\times$  ESTUDIANTE

DONDE (ALEATORIZA.est=ESTUDIANTE.id Y solucoin = "") [nombre, apellidos]

b) Nombre y apellidos de los estudiantes a los que les han asignado más de una pregunta.

DEFINIR ALIAS A1 PARA ALEATORIZA

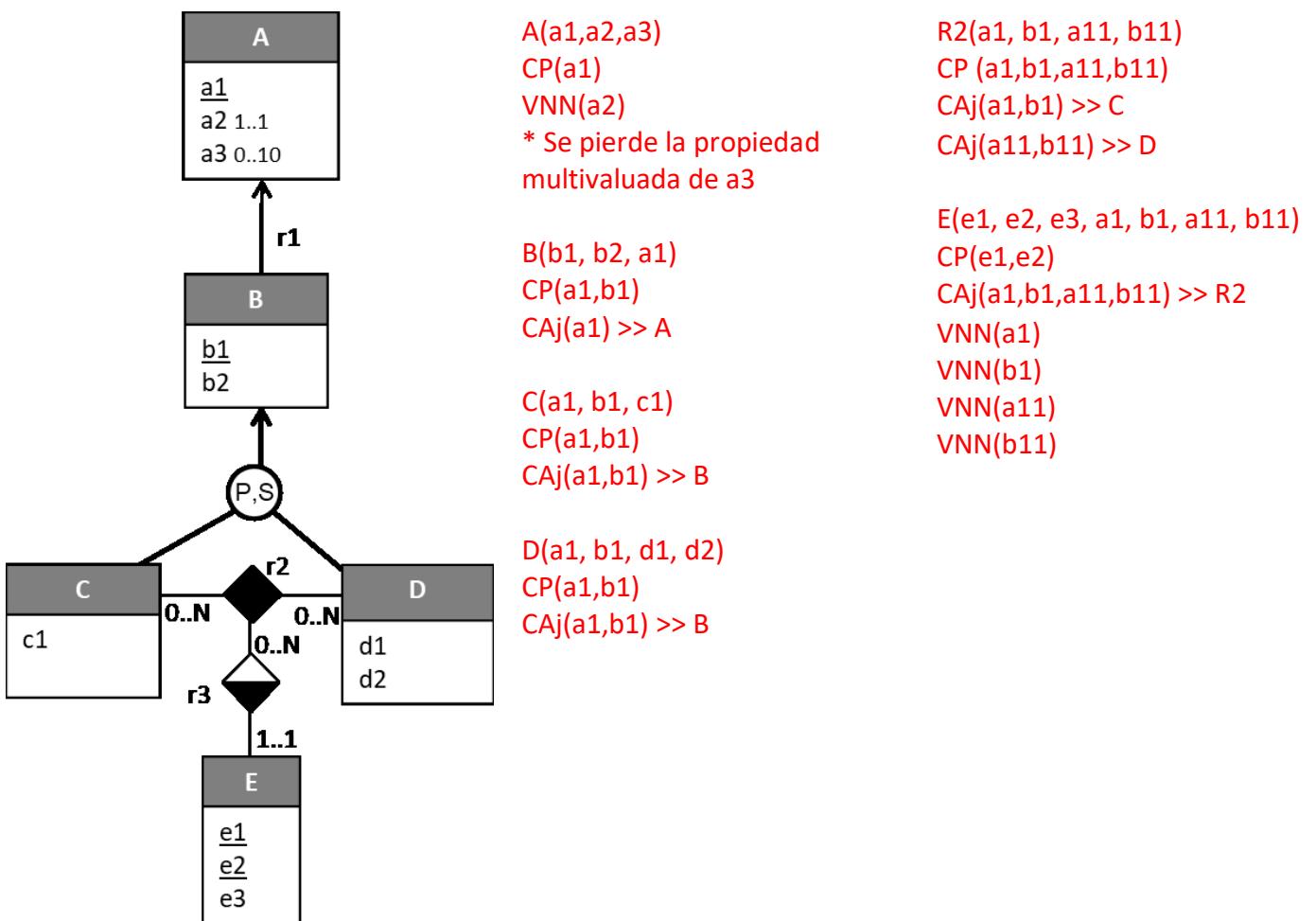
DEFINIR ALIAS A2 PARA ALEATORIZA

$A1 \times A2$  DONDE ( $A1.est = A2.est$  Y ( $A1.cat <> A2.cat$  O  $A1.num <> A2.num$ ))

$\times$  ESTUDIANTE DONDE ESTUDIANTE.id = A1.est

[nombre, apellidos]

- 4) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.



D.N.I.: \_\_\_\_\_ Apellidos y Nombre: \_\_\_\_\_  
 Aula y fecha: \_\_\_\_\_ Tiempo: 60 min Puntuación: 8 (x5 = 40 puntos)

**1) Representa las dependencias funcionales que han dado lugar a estas tablas (0/2=1 o más errores / sin errores)**

De esta tabla, T(email, nombre, cod, serie, pvp, descripción, fecha), tras aplicar todos los pasos necesarios para normalizar hasta FNBC se han obtenido estas otras tablas

**COMPRA (usuario, articulo, fecha) CP (usuario, articulo, fecha)**

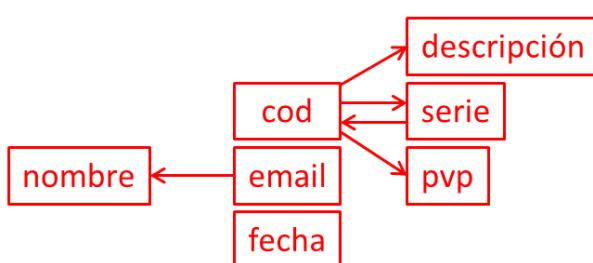
**CAj(usuario) → USUARIO**

**CAj(artículo) → ARTÍCULO**

**USUARIO (email, nombre) CP (email)**

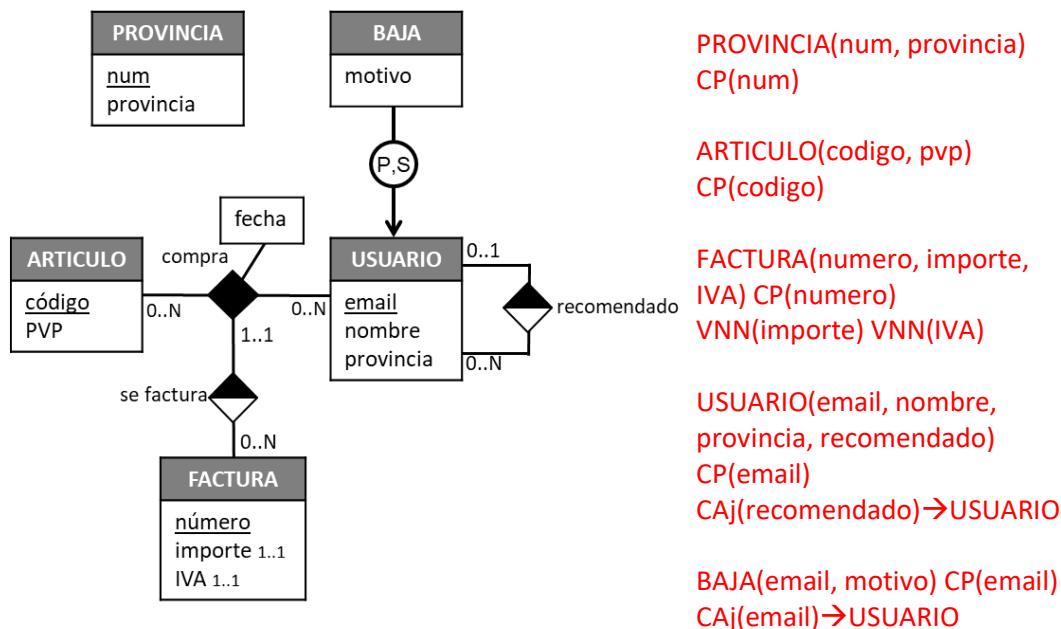
**ARTÍCULO (cod, serie, PVP, descripción) CP (cod) CAAlt (serie)**

¿Qué dependencias funcionales había entre los atributos de T para que el esquema definitivo haya sido el de COMPRA, USUARIO Y ARTÍCULO?



\* los nombres de atributo pueden cambiar

**2) Transforma este esquema entidad-relación en un esquema relacional (0/1/2=más de 2 errores / 2 errores / 1 o sin errores) No es necesario definir dominios para ningún atributo. Cualquier restricción que se pierda en la transformación debes reflejarla como nota al pie.**



PROVINCIA(num, provincia)  
 CP(num)

ARTICULO(código, pvp)  
 CP(código)

FACTURA(numero, importe,  
 IVA) CP(numero)  
 VNN(importe) VNN(IVA)

USUARIO(email, nombre,  
 provincia, recomendado)  
 CP(email)  
 CAj(recomendado)→USUARIO

BAJA(email, motivo) CP(email)  
 CAj(email)→USUARIO

COMPRA(articulo, usuario,  
 factura, fecha)  
 CP(articulo, usuario)  
 CAj(articulo)→ARTICULO  
 CAj(usuario)→USUARIO  
 CAj(factura)→FACTURA  
 VNN(factura)

VNN(importe) VNN(IVA)

### 3) Relaciones (0/1/2=más de 2 errores/2 errores/1 o sin errores)

Se trata de una base de datos de una empresa de distribución en línea. Mantiene datos de usuarios y artículos, y qué compran ellos; los usuarios valoran artículos y recomiendan a amigos.

<b>USUARIO</b> (email,nombre,provincia) <b>CP</b> (email)	<b>COMPRADO</b> (cod, email, num) <b>CP</b> (cod,email,num) <b>CAj</b> (cod) → ARTÍCULO <b>CAj</b> (email,num) → FACTURA	<b>RECOMIENDA</b> (usuario,amigo) <b>CP</b> (usuario) <b>CAlt</b> (amigo) <b>CAj</b> (usuario)→USUARIO <b>CAj</b> (amigo)→USUARIO
<b>ARTÍCULO</b> (cod,PVP,serie) <b>CP</b> (cod)	<b>VALORA</b> (id, email, artículo, stars) <b>CP</b> (id) <b>CAlt</b> (email,artículo) <b>CAj</b> (email)→USUARIO <b>CAj</b> (artículo)→ARTÍCULO <b>VNN</b> (stars)	

Según el esquema anterior, rellena las siguientes expresiones con sus valores correspondientes:

- |                                             |                                             |
|---------------------------------------------|---------------------------------------------|
| a) Card(FACTURA, de) = <b>(1,1)</b>         | b) Card(USUARIO, de) = <b>(0,N)</b>         |
| c) Card(ARTÍCULO, comprado) = <b>(0,N)</b>  | d) Card(FACTURA, comprado) = <b>(0,N)</b>   |
| e) Card(USUARIO, valora) = <b>(0,N)</b>     | f) Card(ARTÍCULO, valora) = <b>(0,N)</b>    |
| g) Card(USUARIO, recomienda) = <b>(0,1)</b> | h) Card(USUARIO, recomienda) = <b>(0,1)</b> |

### 4) Dado el esquema de base de datos anterior, y suponiendo que todos los dominios son iguales, resuelve en álgebra relacional (0/1/2=ninguna/1 correcta/2 correctas)

a) SELECT nombre FROM usuario WHERE email NOT IN (SELECT email FROM factura)

**USUARIO × FACTURA [nombre]**

Solución alternativa:

USUARIO[email]-(FACTURA[email]) × USUARIO[nombre]

b) SELECT id, stars, provincia FROM valora NATURAL JOIN usuario WHERE stars>4

**VALORA × USUARIO donde stars>4 [id,stars,provincia]**

Solución alternativa:

VALORA × USUARIO

donde VALORA.email = USUARIO.email y stars>4

[id,stars,provincia]