

---

## EJERCICIOS CAPÍTULO 8

### Ejercicio 1

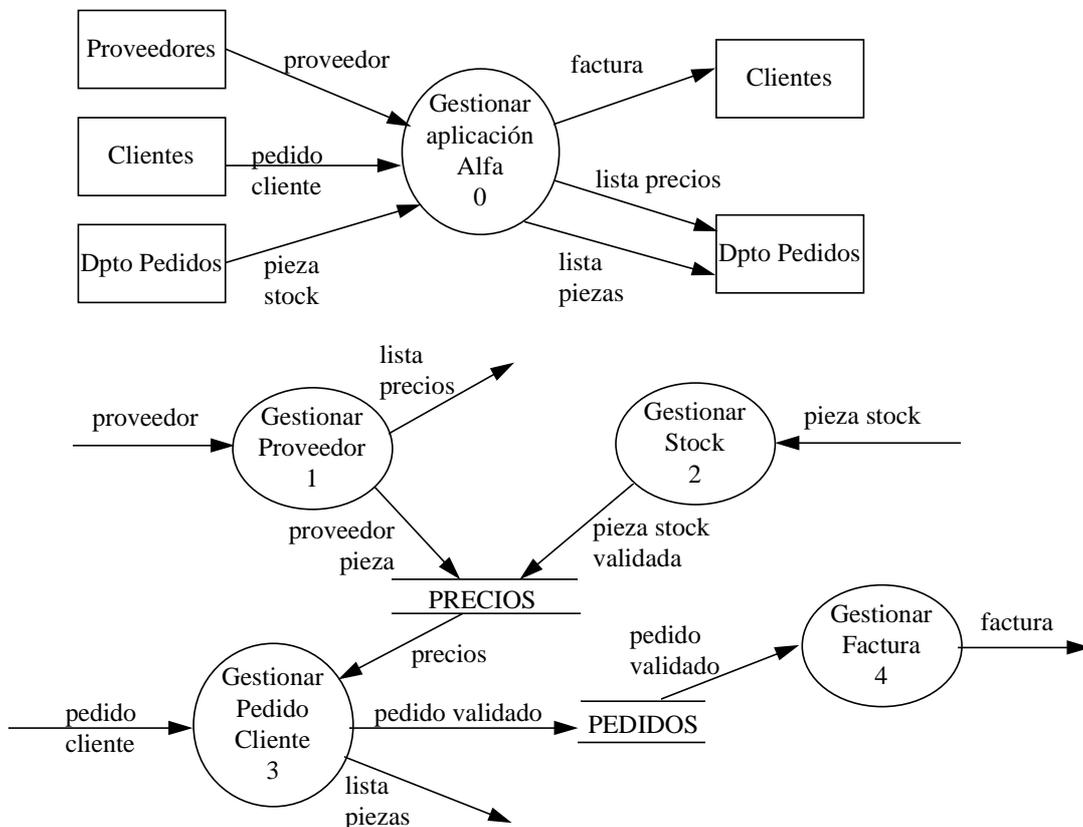
Una empresa compra a una serie de proveedores diferentes piezas que posteriormente venderá a sus clientes, debiendo llevar a cabo el control de almacén (nº de piezas existentes de cada una de ellas).

La aplicación debe gestionar los proveedores, así como las piezas que proporciona cada uno (proveedor y piezas con sus respectivos precios, corresponde al flujo de entrada «proveedor»). Con los proveedores y las piezas que proporciona cada uno de ellos, se genera una lista de precios que se corresponde con los precios que consideremos mejores para cada una de las piezas que se puedan proporcionar al cliente (como criterio de selección se encuentra entre otros la marca de la pieza).

El control del almacén, es decir, las cantidades que tenemos de las diferentes piezas que hemos pedido a los proveedores (flujo de datos de «pieza stock»), determinará si el pedido realizado por el cliente («pedido cliente») se puede satisfacer completamente o no, según tengamos o no las piezas pedidas (generando en el caso de no tener dichas piezas un listado de ellas, «lista piezas»).

Cuando el pedido se entrega al cliente, se genera la factura correspondiente.

Cada una de estas funciones (en el DFD 1, 2, 3 y 4) puede realizarse en cualquier momento, independientemente de las demás funciones. Se pide dibujar el diagrama de estructuras correspondiente indicando si la característica principal del DFD es de transformación o transacción.



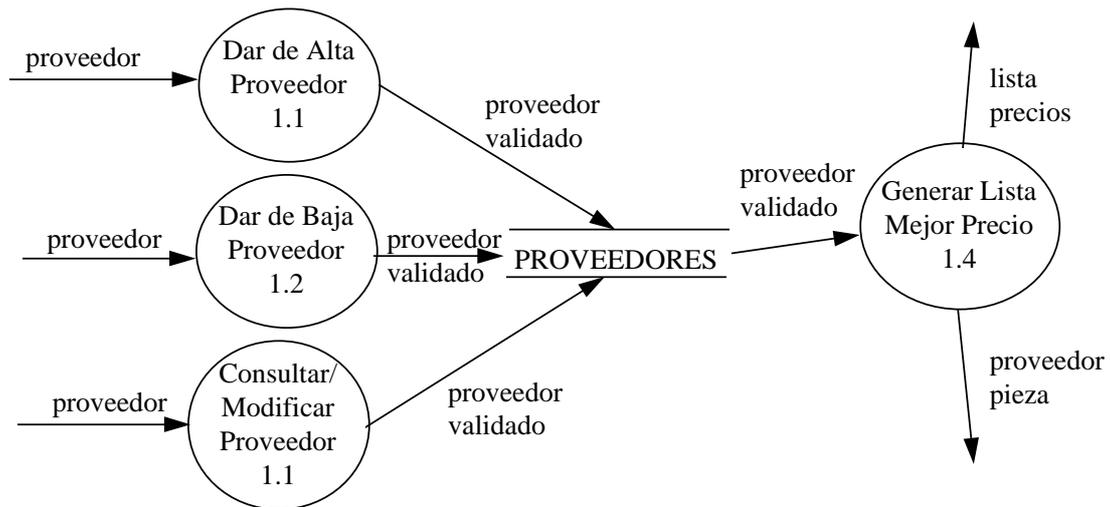


Figura 8.4 DFD del ejercicio 1

## Ejercicio 2

Se desea automatizar la gestión de un VideoClub. El funcionamiento del sistema es el siguiente:

- Por un lado, es necesario tratar a los proveedores y, por otro, es necesario tratar a los clientes.
- El tratamiento de los proveedores incluye, en cualquier instante y de forma independiente, la realización del registro de los catálogos de películas y la generación de pedidos de películas al proveedor.
- Cuando llega un cliente al videoclub, éste solicita el tipo de gestión que quiere realizar, es decir, alquiler, reserva o devolución de película. Cuando quiere realizar un alquiler de una película, el proceso, con la información del alquiler, comprobará si existe stock suficiente de esa película, así como reserva. En caso de ser satisfactorias estas comprobaciones (existe la película y no está reservada por otro cliente), se disminuirá el stock de esa película y se registrará el alquiler generando un comprobante de alquiler para el cliente. Cuando quiere realizar una devolución de una película, lo que se hace es comprobar que la película estaba alquilada por él y aumentar el stock de esa película. Cuando quiere reservar una película, se registra la reserva de la película.

No se tiene en cuenta el tratamiento de los errores que pudieran ocurrir.

Se pide: realizar el diagrama de estructura correspondiente, indicando si se trata de una transformación (decir cuál es su centro) o de una transacción.

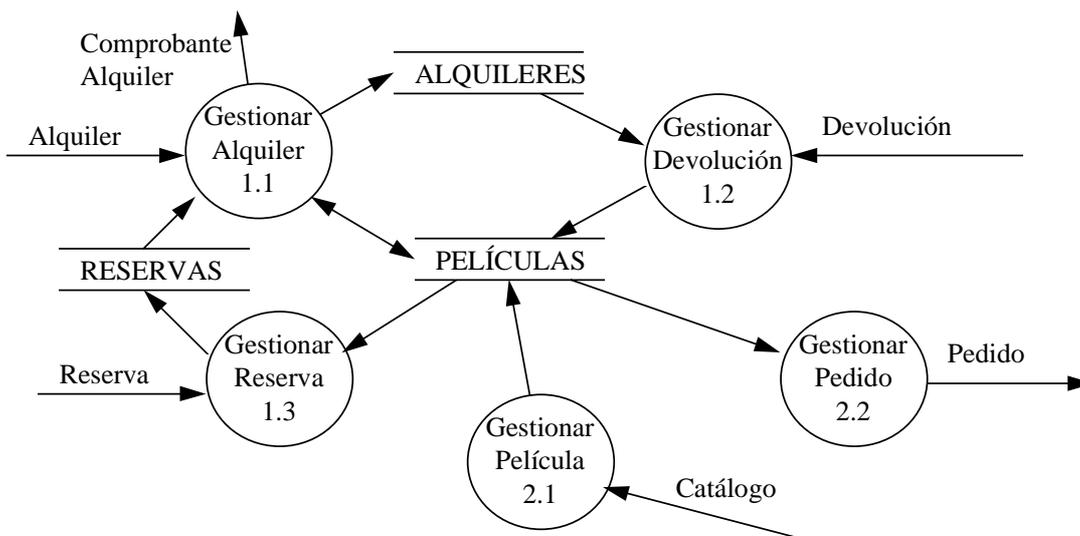
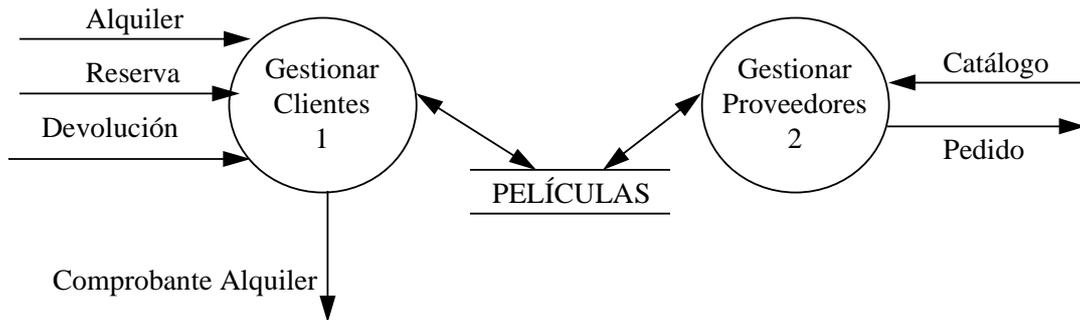
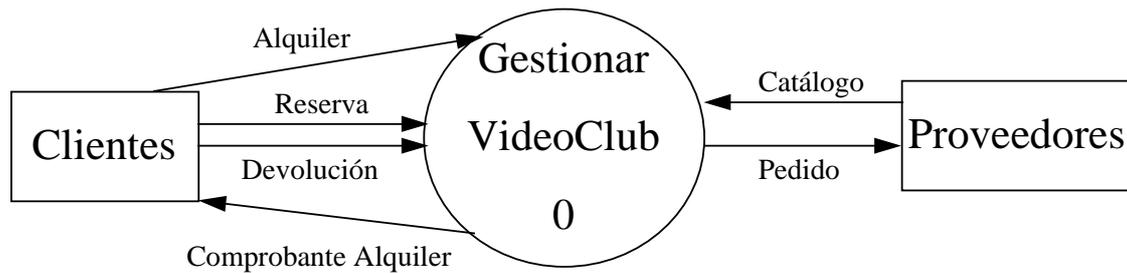


Figura 8.42. DFD del ejercicio 2

### Ejercicio 3

Dado el esquema E/R de la figura 8.43, en el que entidades tienen los siguientes atributos:

JUGADOR (DNI, NOMBRE\_J, FECHA\_NACIMIENTO, SUELDO, NACIONALIDAD)

EQUIPO (NOMBRE\_EQ, CIUDAD, PRESIDENTE)

PARTIDO (CÓDIGO, FECHA, RESULTADO)

Se pide:

1. Pasar al modelo relacional, aplicando las reglas usuales de derivación.

2. Describir el esquema resultante en el lenguaje SQL, o en cualquier otro que conozca.
3. ¿Sugeriría alguna modificación en el esquema relacional obtenido? En caso afirmativo, ¿podría formular una nueva regla de derivación?

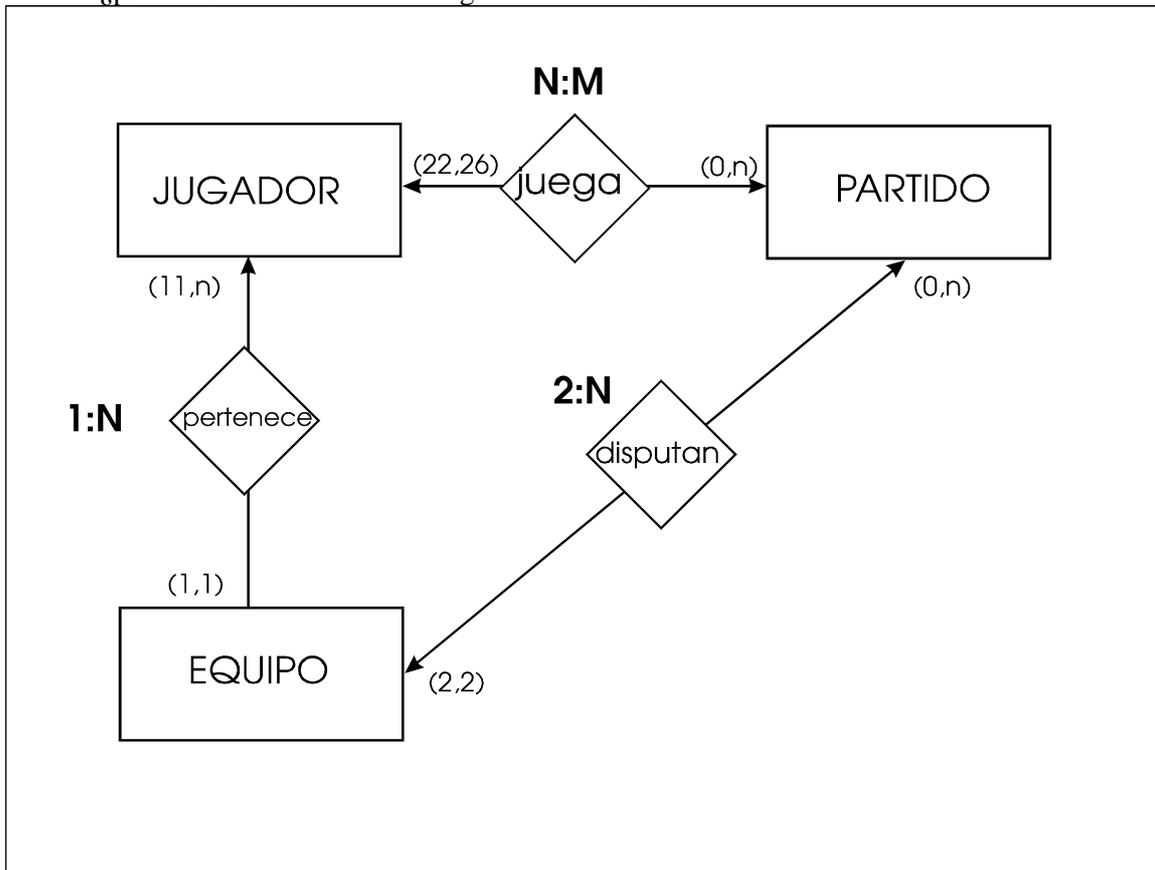


Figura 8.43. Esquema E/R del ejercicio 3

#### Ejercicio 4

La seguridad social desea conocer los pacientes (DNI, NOMBRE, DIRECCIÓN) que han sido atendidos en sus hospitales (CÓD\_HOSP) y el doctor que los atiende (CÓD\_DOC). Suponiendo que un doctor sólo puede atender en un hospital y que, aunque un paciente puede ser atendido en varios hospitales, en cada uno de ellos sólo le atiende un doctor, determinar las dependencias funcionales y la forma normal de la relación:

R (DNI, NOMBRE, DIRECCIÓN, CÓD\_HOSP, CÓD\_DOC)

Llevar esta relación hasta la forma normal que se crea más conveniente explicando los motivos. Especificar todos los supuestos adicionales que se han tenido en cuenta.

#### Ejercicio 5

Llevar las siguientes relaciones hasta la forma normal que se crea más conveniente, explicando los motivos. Especificar todos los supuestos adicionales que se han tenido en cuenta.

1. VIDEOCLUB (TÍTULO, AÑO, NOMBRE\_ACTOR, NOMBRE\_DIRECTOR, NACIONALIDAD\_DIRECTOR)

Se supone que una película sólo tiene un director, pero un director puede dirigir más de una.

## 2. MENÚ (PERSONA, CLASE, PLATO)

Se supone que existen tres clases (primero, segundo y postre), y que cada persona sólo come un plato de cada clase. Por otra parte, cada plato (por ejemplo, «sopa») sólo pertenece a una clase (primero).

### Ejercicio 6

Dado el diagrama E/R de la figura 8.44, se pide pasarlo al modelo relacional, proponiendo varias alternativas (en lo que respecta a la jerarquía de generalización), valorando las ventajas e inconvenientes de cada una.

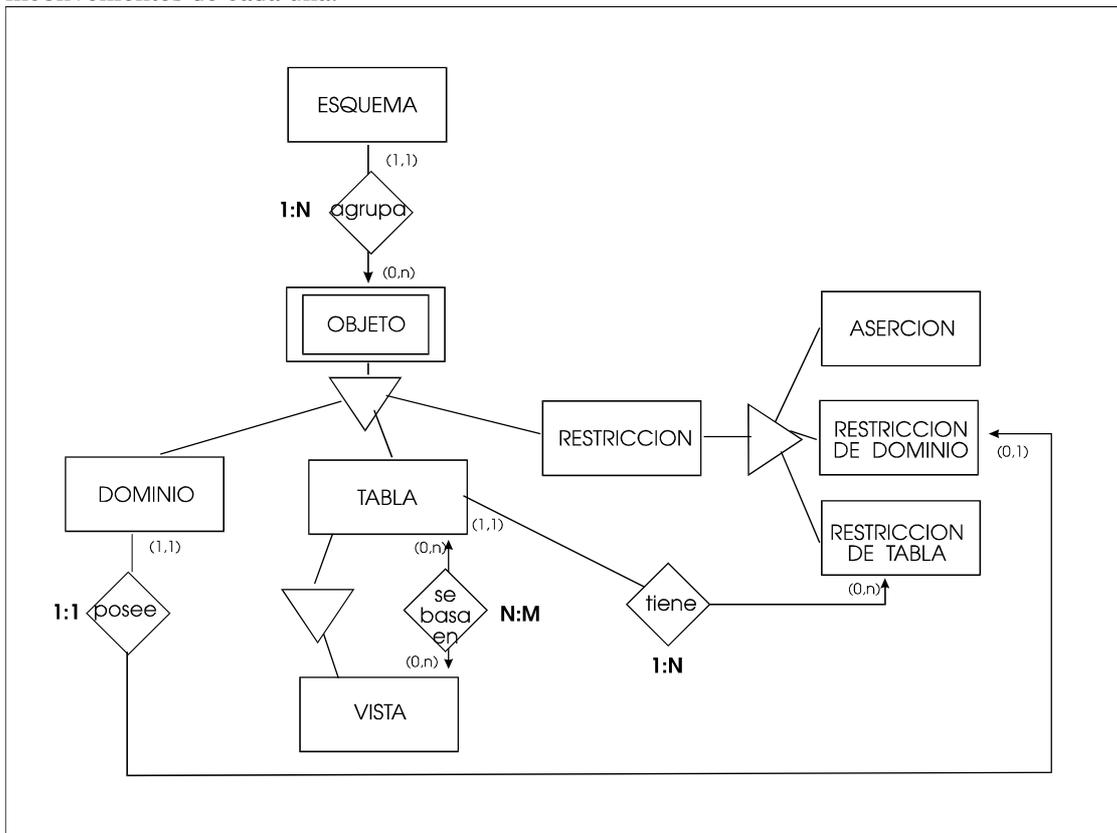


Figura 8.44. Diagrama E/R del ejercicio 6

---

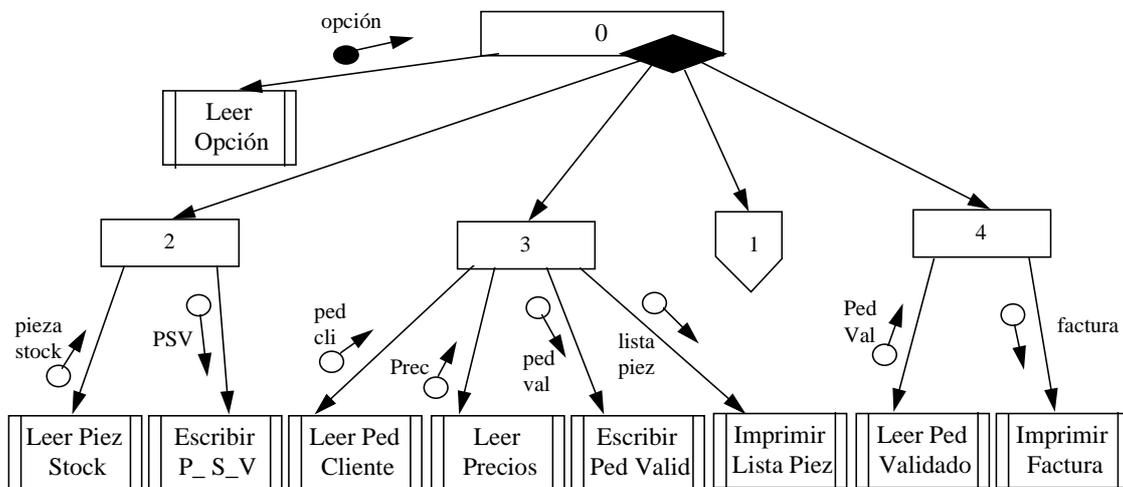
## SOLUCIONES A LOS EJERCICIOS

### Ejercicio 1

En este ejercicio, el enunciado refleja una transacción en el primer nivel, ya que nos dice que cualquiera de las funciones del diagrama principal «*puede realizarse en cualquier momento, independientemente de las demás funciones*». Esto quiere decir que en un instante determinado se puede realizar cualquiera de las funciones indicadas sin que previamente se necesite haber realizado alguna de ellas en particular.

En este ejercicio no se ha contemplado el tratamiento de errores. Así, por ejemplo, no se podrá realizar una factura si no existe ningún pedido. Aunque esto pudiera indicar que sería necesario realizar primero el proceso o módulo 3 y posteriormente el 4, en la realidad no es así, ya que nosotros podemos acceder primero al módulo 4 y, en el caso de que no hubiese ningún pedido no se realizaría la factura, apareciendo el mensaje de error o de aviso que nos indicaría que no existe ningún pedido al que se le pueda hacer factura. Tal como hemos visto primero puedo procesar el módulo 4 y luego el 3 o viceversa, es decir el orden depende de la selección que haga la persona que trabaje con el programa.

Así mismo, existe otra transacción en el módulo 1, ya que no podremos dar de alta, de baja o consultar/modificar a un proveedor de forma simultánea. El proceso 1.4 puede ser contemplado por algunos lectores también como parte de la transacción, sin embargo, en la solución hemos decidido que sea un proceso que se hace siempre, independientemente de la opción que se haya seleccionado.



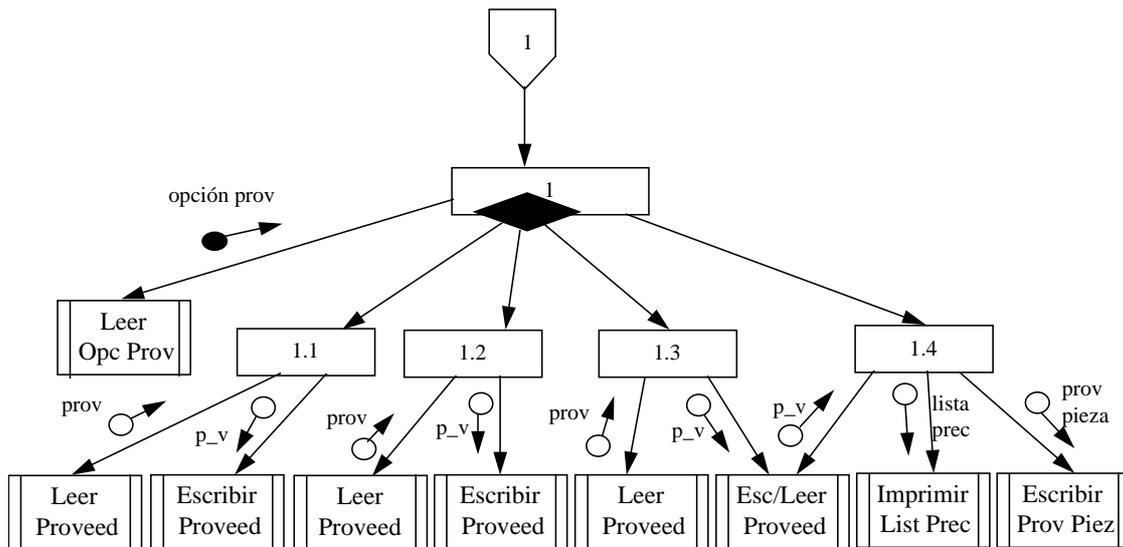


Figura 8.45. Solución Ejercicio 1

## Ejercicio 2

En este ejercicio, el enunciado tiene ocultos los centros de transacción existentes en la solución propuesta. El primer centro de transacción, módulo 0, se corresponde con el párrafo «por un lado, es necesario tratar a los proveedores y, por otro, es necesario tratar a los clientes». Evidentemente, con un conocimiento más profundo del sistema se llegaría a esta conclusión, dado que está frase pudiera ser ambigua para algunos lectores.

La segunda transacción se produce con los clientes, módulo 1, y es evidente que un cliente no puede alquilar, devolver o reservar una película, simultáneamente sino que si quiere realizar estas tres acciones tendrá que hacerlas una **depués** de otra.

La tercera transacción, módulo 2, se produce porque el enunciado dice que tanto el registro de catálogos de películas como la generación de pedidos de películas se realizan de forma independiente, es decir, que ninguna función necesita de la otra para ejecutarse.

En este ejercicio no se ha contemplado el tratamiento de errores, que, evidentemente, debe tener cualquier aplicación que se haga. Este tratamiento de errores debería contemplar, por ejemplo, que no se puede devolver una película si no se ha alquilado previamente. Esto no quiere decir que al ejecutar el programa, con la solución propuesta, no se pueda realizar primero el módulo de devoluciones y luego cualquier otra opción, ya que si primero se ejecuta el módulo de devoluciones y no hay alquileres, daría el correspondiente mensaje de error.

Tanto esta solución como la del ejercicio anterior reflejan la programación de los programas basados en windows, donde se pueden tener abiertas muchas ventanas en paralelo, aunque realmente sólo se trabaja con una. La forma de expresar este tipo de programación es mediante transacciones.

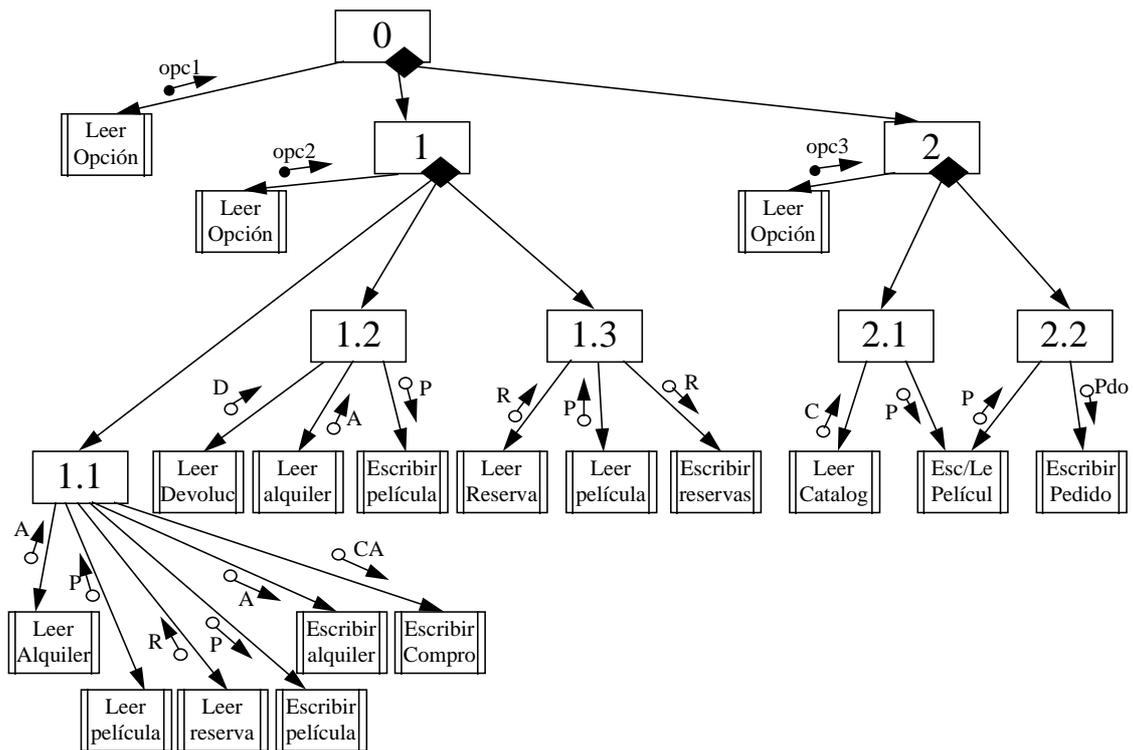


Figura 8.46. Solución ejercicio 2

### Ejercicio 3

1. Aplicando las reglas usuales, toda entidad se transforma en una relación, toda interrelación N:M se transforma en una relación y toda interrelación 1:N se traduce en el mecanismo de propagación de clave, por lo que el esquema relacional resultante es:

JUGADOR (DNI, NOMBRE\_J, FECHA\_NACIMIENTO, SUELDO, NACIONALIDAD, NOMBRE\_EQ)  
 EQUIPO (NOMBRE\_EQ, CIUDAD, PRESIDENTE)  
 PARTIDO (CÓDIGO, FECHA, RESULTADO)  
 JUEGA (DNI, CÓDIGO, PUESTO)  
 DISPUTAN (CÓDIGO, NOMBRE\_EQ)

2. La descripción en lenguaje SQL quedaría así:

```

CREATE DOMAIN NOMBRES AS VARCHAR (25)
CREATE DOMAIN DINERO AS INTEGER (8)
CHECK (VALUE >=0)
CREATE DOMAIN CODIGO AS CHAR (6)

CREATE TABLE JUGADOR
(
  DNI          CHAR (10),
  NOMBRE_J    NOMBRES NOT NULL,
  FECHA_NAC   DATE NOT NULL,
  SUELDO      DINERO,
  NACIONALIDAD NOMBRES NOT NULL,
  NOMBRE_EQ   NOMBRES NOT NULL,

  CONSTRAINT CP_JUGADOR CHECK
  PRIMARY KEY DNI,

```

---

```
CONSTRAINT JUGADOR_EQUIPO
FOREIGN KEY NOMBRE_EQ REFERENCES EQUIPO
ON DELETE RESTRICT
ON UPDATE CASCADE
)
```

```
CREATE TABLE EQUIPO
(
NOMBRE_EQ          NOMBRES,
CIUDAD            NOMBRES  NOT NULL,
PRESIDENTE        NOMBRES  NOT NULL
PRIMARY KEY NOMBRE_EQ
)
....
```

Además, para reflejar toda la semántica del esquema, se deberían añadir restricciones y aserciones que controlaran las cardinalidades mínimas y máximas distintas de 0, 1, n.

3. Se podría sugerir transformar la relación, ya que la cardinalidad mínima y máxima de la entidad equipo es dos, de modo que resultaría:

```
DISPUTAN (CÓDIGO_PARTIDO, EQUIPO_LOCAL, EQUIPO_VISITANTE)
```

como su clave primaria es la misma que partido, se podrían fusionar ambas relaciones en una sola:

```
PARTIDO (CÓDIGO, FECHA, EQUIPO_LOCAL, EQUIPO_VISITANTE,
RESULTADO)
```

Se podría incluir una nueva regla a considerar al pasar interrelaciones de cardinalidades máximas distintas de 1 o n, que consistiría en propagar tantas veces la clave de una entidad como cardinalidad tuviese en la otra entidad (en este caso, se propagaría la clave de equipo 2 veces a partido). Esta regla sería útil para casos en los que la cardinalidad es un número pequeño y si tanto la máxima como la mínima son iguales, se garantiza evitar valores nulos.

## Ejercicio 4

Si suponemos que los nombres de los pacientes no son únicos y que en una misma dirección podrían vivir varios pacientes, las dependencias funcionales de la relación r son:

```
DNI → NOMBRE
DNI → DIRECCIÓN
CÓD_DOC → CÓD_HOS
DNI, CÓD_HOS → CÓD_DOC
```

Las claves de la relación serían {DNI, CÓD\_HOS} y {DNI, CÓD\_DOC}, al existir atributos no principales (nombre, dirección) que dependen de parte de las claves pero no de las claves completas, es decir, al existir dependencias funcionales no completas, la relación r no se encuentra en 2FN. Para llevarla a 2FN la descomponemos en proyecciones independientes según el principio de Rissanen:

```
ENFERMO (DNI, NOMBRE, DIRECCIÓN)
R' (DNI, CÓD_DOC, CÓD_HOS)
```

La relación enfermo se encuentra en FNBC, ya que todo determinante es clave (DNI).

---

La relación  $r'$  se encuentra en 3FN, ya que no existen atributos no principales, pero no está en FNBC, ya que no todo determinante es clave, pues  $CÓD\_DOC$  es determinante pero no es clave. Las claves, que se solapan, son  $\{DNI, CÓD\_HOS\}$  Y  $\{DNI, CÓD\_DOC\}$ ,

Para llevar la relación  $r'$  a la FNBC la podemos descomponer de tres formas distintas, la mejor es:

R1 ( $CÓD\_DOC, CÓD\_HOS$ )  
R2 ( $DNI, CÓD\_DOC$ )

Ya que en cualquier otro caso se pierde información (apareciendo tuplas espurias). A pesar de todo se pierde la dependencia funcional  $DNI, COD\_HOS \rightarrow COD\_DOC$ , por lo que podría ser conveniente dejarla en 3FN.

## Ejercicio 5

1.- Si suponemos que los títulos de las películas no se pueden repetir, que un actor puede hacer varios papeles, y que en una película actúan varios actores.

TÍTULO  $\rightarrow$  NOMBRE\_DIRECTOR  
TÍTULO  $\rightarrow$  AÑO  
NOMBRE\_DIRECTOR  $\rightarrow$  NACIONALIDAD\_DIRECTOR

La clave de la relación será  $\{TÍTULO, NOMBRE\_ACTOR\}$ , por lo que al existir atributos no principales (como, por ejemplo, AÑO o NOMBRE\_DIRECTOR) que dependen de parte de la clave pero no de la clave completa, la relación VIDEOCLUB no se encuentra en 2FN.

Para llevarla a 2FN la descomponemos en proyecciones independientes según el principio de Rissanen:

(TÍTULO, ACTOR)  
(TÍTULO, NOMBRE\_DIRECTOR, AÑO, NACIONALIDAD\_DIRECTOR)

La primer relación se encuentra ya en FNBC, mientras que la segunda no se encuentra en 3FN, ya que el atributo NACIONALIDAD\_DIRECTOR depende transitivamente de TÍTULO, a través de NOMBRE\_DIRECTOR. Para llevarla a 3FN la descomponemos en proyecciones independientes:

(TÍTULO, AÑO, NOMBRE\_DIRECTOR)  
(NOMBRE\_DIRECTOR, NACIONALIDAD\_DIRECTOR)

que ya se encuentran en FNBC, al ser todo determinante una clave.

2.- Las dependencias funcionales son las siguientes:

(PERSONA, CLASE)  $\rightarrow$  PLATO  
PLATO  $\rightarrow$  CLASE

Las claves de la relación serían:  $\{PERSONA, CLASE\}$  y  $\{PERSONA, PLATO\}$ . Al no existir atributos no principales la relación se encuentra en 3FN, pero no está en FNBC, ya que no todo determinante es clave. En efecto, PLATO es un determinante, pero no es clave (forma parte de una, pero no ES clave).

Para llevar esta relación a FNBC la podríamos descomponer de las siguientes maneras:

a) (CLIENTE, CLASE)  
(PLATO, CLASE)

Descomposición en la que perdemos información (aparecen tuplas espurias).

- 
- b) (CLIENTE, PLATO)  
(CLIENTE, CLASE)

Perdemos las dos dependencias funcionales.

- c) (CLIENTE, PLATO)  
(PLATO, CLASE)

Se pierde la dependencia (CLIENTE, CLASE) -> PLATO, por lo que en este caso no sería demasiado conveniente llevar la relación hasta la FNBC.

## Ejercicio 6

Una opción posible sería transformar cada super-tipo y sub-tipo de entidad en una relación:

ESQUEMA (NOMBRE\_ESQ, AUTORIZACIÓN, ...)  
OBJETO (NOMBRE\_ESQ, NOMBRE\_OBJETO, ...)  
DOMINIO (NOMBRE\_ESQ, NOMBRE\_OBJETO, TIPO DE DATOS, ...)  
TABLA (NOMBRE\_ESQ, NOMBRE\_OBJETO, NÚMERO\_FILAS, ...)  
RESTRICCIÓN (NOMBRE\_ESQ, NOMBRE\_OBJETO, PREDICADO, ...)  
VISTA (NOMBRE\_ESQ, NOMBRE\_OBJETO, CHECK\_OPTION, ACTUALIZABLE, ...)  
ASERCIÓN (NOMBRE\_ESQ, NOMBRE\_OBJETO, ...)  
RESTRICCIÓN\_DOMINIO (NOMBRE\_ESQ, NOMBRE\_OBJETO, NOMBRE\_ESQ\_DOMINIO, NOMBRE\_OBJETO\_DOMINIO ...)  
RESTRICCIÓN\_TABLA (NOMBRE\_ESQ, NOMBRE\_OBJETO, NOMBRE\_ESQ\_TABLA, NOMBRE\_OBJETO\_TABLA ...)  
SE\_BASA\_EN (NOMBRE\_ESQ TABLA, NOMBRE\_OBJETO TABLA, NOMBRE\_ESQ VISTA, NOMBRE\_OBJETO VISTA )

Otra solución consistiría en transformar las jerarquías en una sola relación, pero en este caso tendríamos:

ESQUEMA (NOMBRE\_ESQ, AUTORIZACIÓN, ...)  
OBJETO (NOMBRE\_ESQ, NOMBRE\_OBJETO, CLASE, TIPO DE DATOS, NÚMERO\_FILAS, PREDICADO, CHECK\_OPTION, ACTUALIZABLE, NOMBRE\_ESQ\_DOMINIO, NOMBRE\_OBJETO\_DOMINIO, NOMBRE\_ESQ\_TABLA, NOMBRE\_OBJETO\_TABLA ...)  
SE\_BASA\_EN (NOMBRE\_ESQ TABLA, NOMBRE\_OBJETO TABLA, NOMBRE\_ESQ VISTA, NOMBRE\_OBJETO VISTA )

Esta segunda opción presenta la ventaja de que es más eficiente ante ciertas consultas, pero la relación OBJETO presenta una gran cantidad de valores nulos, además de perder semántica respecto a la opción anterior.

Una solución más adecuada podría ser la siguiente:

ESQUEMA (NOMBRE\_ESQ, AUTORIZACIÓN, ...)  
DOMINIO (NOMBRE\_ESQ, NOMBRE\_OBJETO, TIPO DE DATOS, ...)  
TABLA (NOMBRE\_ESQ, NOMBRE\_OBJETO, NÚMERO\_FILAS, ...)  
RESTRICCIÓN (NOMBRE\_ESQ, NOMBRE\_OBJETO, PREDICADO, ...)  
VISTA (NOMBRE\_ESQ, NOMBRE\_OBJETO, CHECK\_OPTION, ACTUALIZABLE, ...)  
ASERCIÓN (NOMBRE\_ESQ, NOMBRE\_OBJETO, ...)  
RESTRICCIÓN\_DOMINIO (NOMBRE\_ESQ, NOMBRE\_OBJETO, NOMBRE\_ESQ\_DOMINIO, NOMBRE\_OBJETO\_DOMINIO ...)  
RESTRICCIÓN\_TABLA (NOMBRE\_ESQ, NOMBRE\_OBJETO, NOMBRE\_ESQ\_TABLA, NOMBRE\_OBJETO\_TABLA ...)

---

SE\_BASA\_EN (NOMBRE ESQ TABLA, NOMBRE OBJETO TABLA,  
NOMBRE ESQ VISTA, NOMBRE OBJETO VISTA )

en la que se aplica para la entidad OBJETO y sus sub-tipos directos la transformación consistente en eliminar el super-tipo y pasar las características comunes a los sub-tipos.