2.5. Transformación de las extensiones del modelo E-R

2.5.1. Especialización

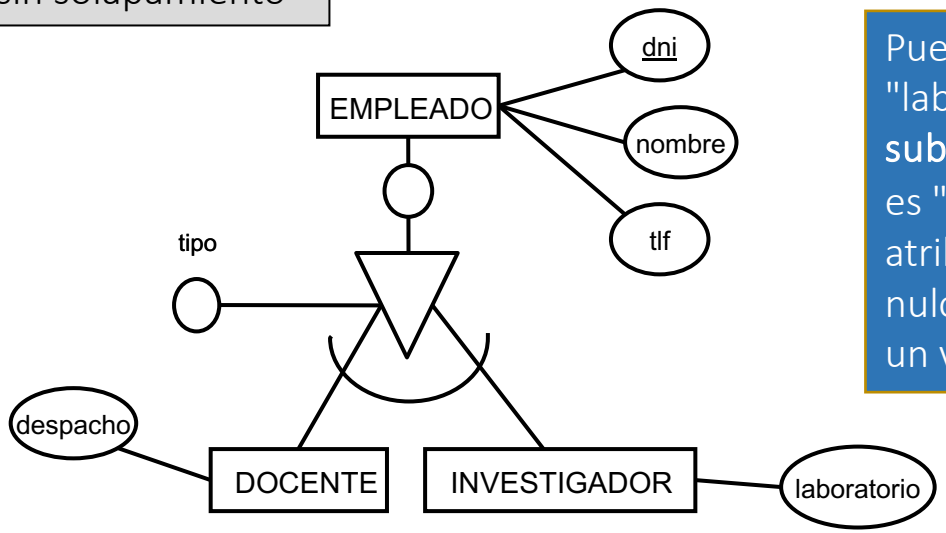
- Existen diferentes opciones para transformar la especialización

A. **Englobar todos los atributos de la entidad y sus subtipos en una sola tabla**
   - Esta solución es conveniente cuando los subtipos se diferencian en muy pocos atributos y las relaciones que los asocian con el resto de las entidades son las mismas para todos los subtipos
   - Para saber a qué subtipo pertenece una tupla podemos consultar la propia información de la tabla. Sin embargo, en algunas situaciones, debemos añadir el atributo discriminante para conocer la pertenencia a los subtipos

Crear una única tabla sin incluir el atributo discriminante

- Para poder realizar esta solución, es necesario tener algún atributo obligatorio específico en cada uno de los subtipos
- Si se cumple esa condición, la solución es válida para cualquier tipo de especialización (total/parcial o con/sin solapamiento)

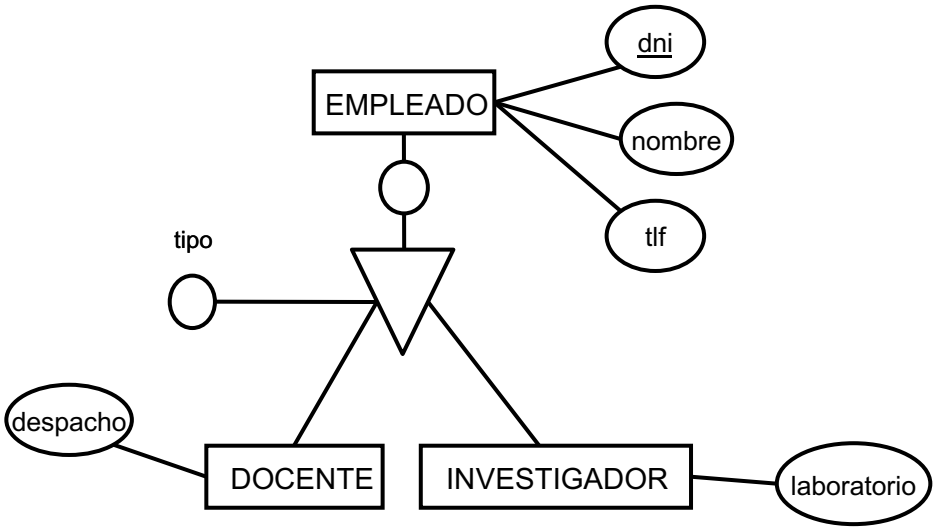
Total sin solapamiento



Puesto que los atributos "despacho" y "laboratorio" son **obligatorios en los subtipos**, para saber si un empleado es "docente", podemos consultar si el atributo "laboratorio" tiene un valor nulo o el atributo "despacho" tiene un valor no nulo

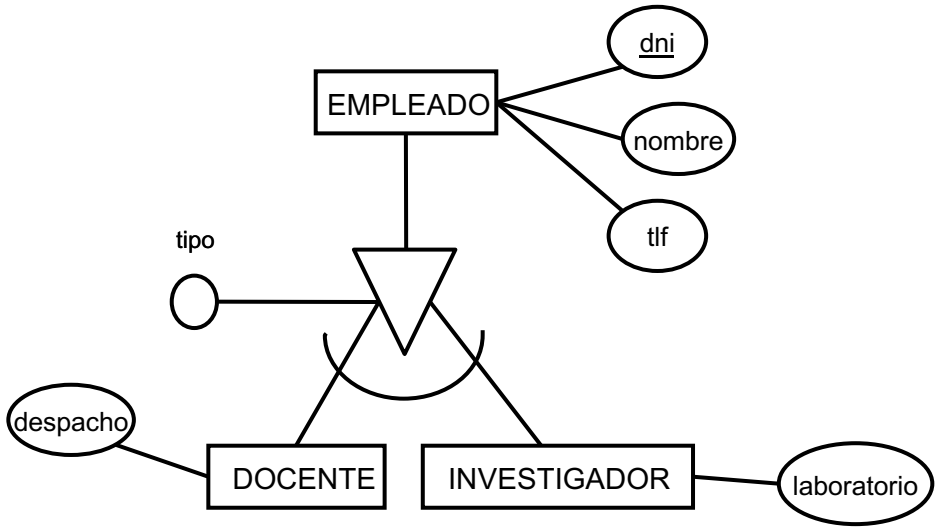
EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf
RESTRICCIÓN Docente_o_Investigador
CHECK ((laboratorio IS NULL AND despacho IS NOT NULL)
OR (despacho IS NULL AND laboratorio IS NOT NULL))

Total con solapamiento



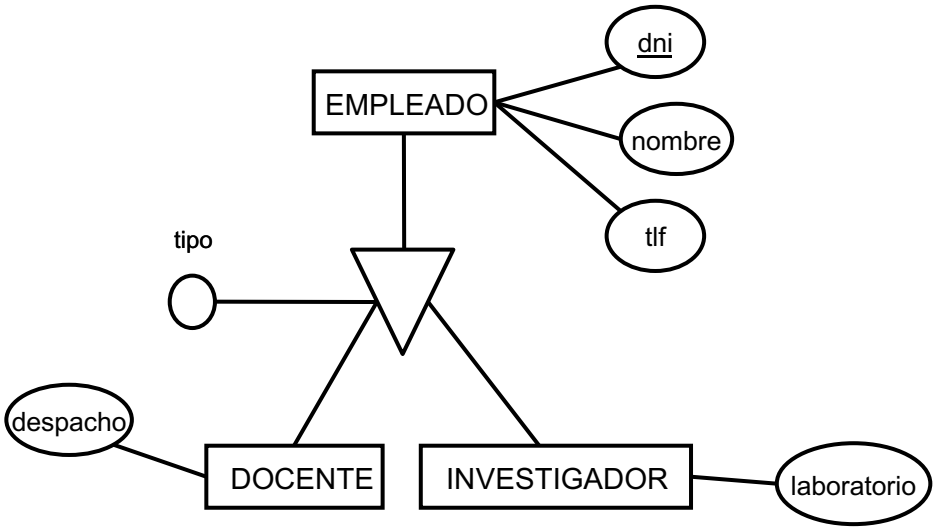
EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf
RESTRICCIÓN Docente_y/o_Investigador
CHECK (laboratorio IS NOT NULL OR despacho IS NOT NULL)

Parcial sin solapamiento



EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf
RESTRICCIÓN Docente_o_Investigador_u_Otro
CHECK ((laboratorio IS NULL AND despacho IS NOT NULL)
OR (despacho IS NULL AND laboratorio IS NOT NULL)
OR (despacho IS NULL AND laboratorio IS NULL))

Parcial con solapamiento

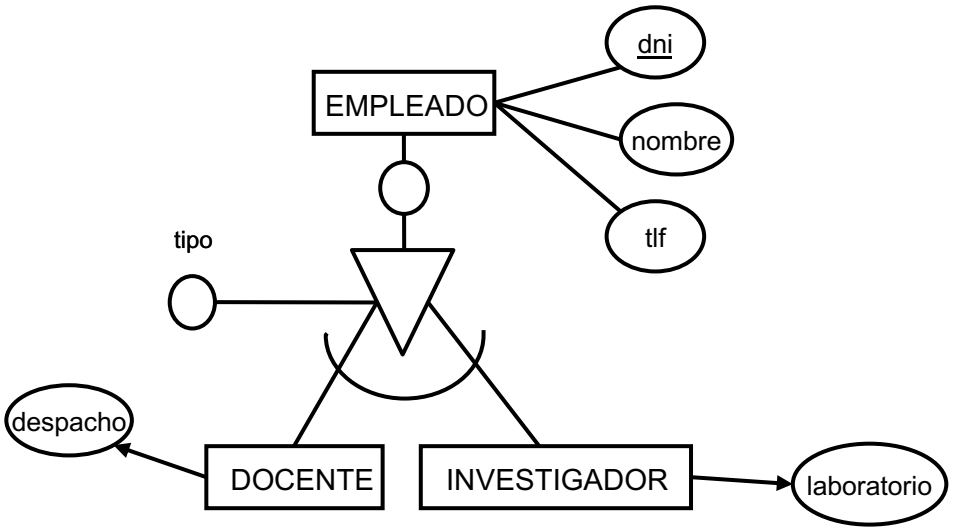


EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf

Crear una única tabla incluyendo el atributo discriminante

- Cuando no hay ningún atributo específico en los subtipos que sea obligatorio, es necesario incluir el atributo discriminante para poder saber a qué subtipo pertenece una tupla
- Esta solución solo es válida para las especializaciones sin solapamiento

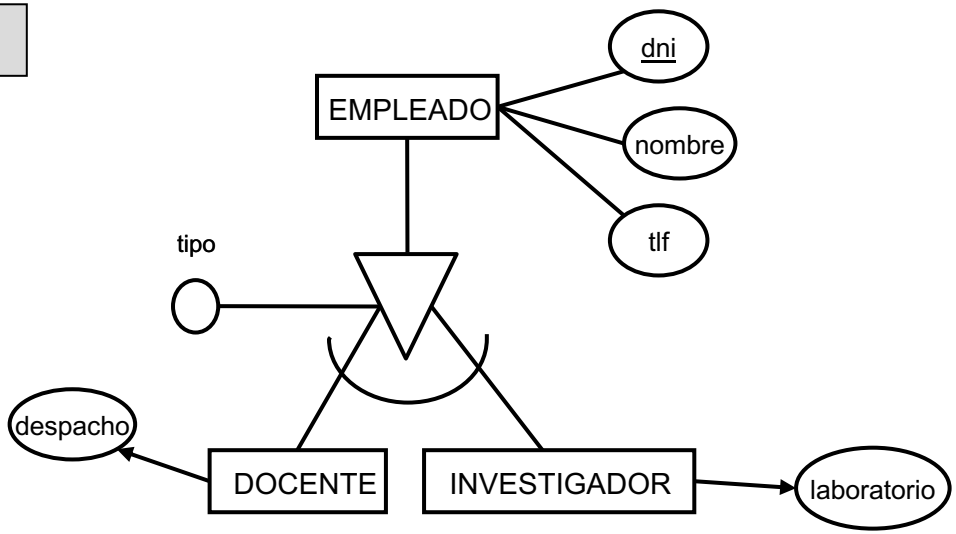
Total sin solapamiento



El atributo discriminante no debe admitir valores nulos

EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho, tipo)
CP: dni
VNN: nombre, tlf, tipo
RESTRICCIÓN Docente_o_Investigador
CHECK ((tipo = 'docente' AND laboratorio IS NULL)
OR (tipo = 'investigador' AND despacho IS NULL))

Parcial sin solapamiento



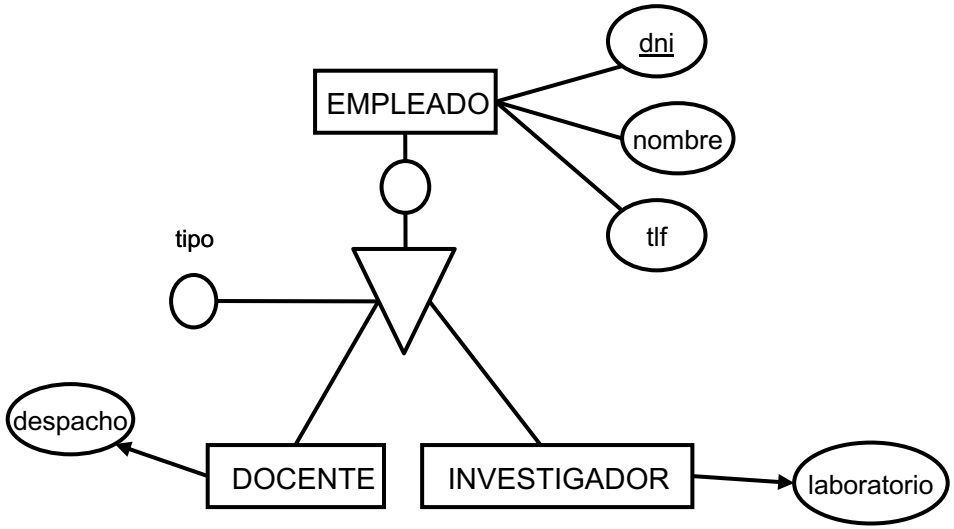
El atributo discriminante debe admitir valores nulos, indicando que no pertenece a ningún subtipo

EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho, tipo)
CP: dni
VNN: nombre, tlf
RESTRICCIÓN Docente_o_Investigador_u_Otro
CHECK ((tipo = 'docente' AND laboratorio IS NULL)
OR (tipo = 'investigador' AND despacho IS NULL)
OR (tipo IS NULL AND despacho IS NULL AND laboratorio IS NULL))

- En las especializaciones con solapamiento en las que haya que incluir el atributo discriminante, es necesario crear una nueva tabla para poder almacenar las ocurrencias que pertenezcan a varios subtipos

Total con solapamiento

Habrá que diseñar los disparadores necesarios para mantener la consistencia entre la información de ambas tablas

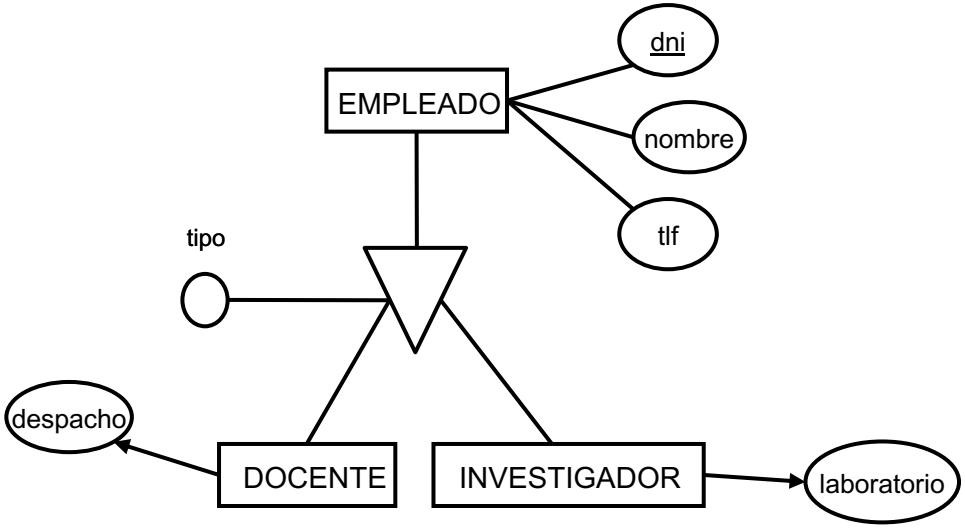


EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf

CATEGORIA (dniEmpleado, tipo)
CP: (dniEmpleado, tipo)
CAj: dniEmpleado → EMPLEADO(dni)

Parcial con solapamiento

Habrá que diseñar los disparadores necesarios para mantener la consistencia entre la información de ambas tablas



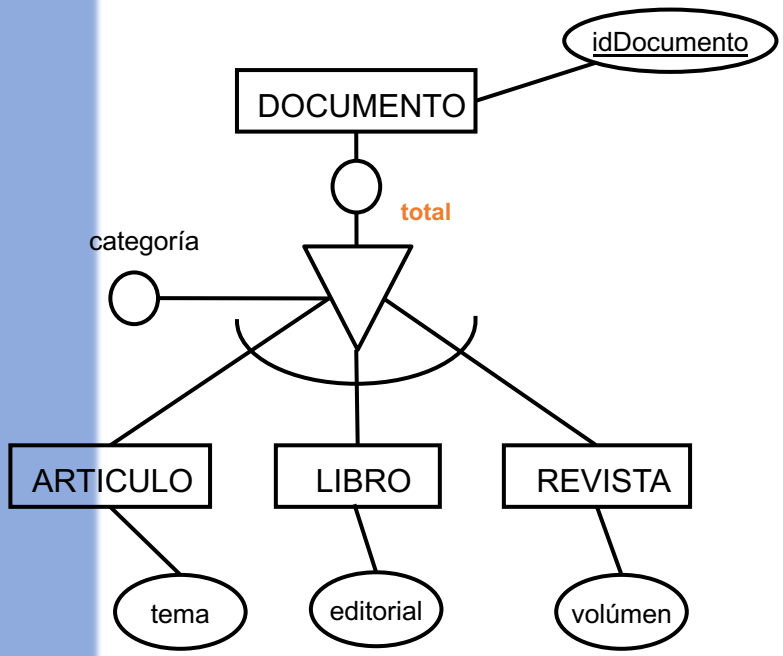
EMPLEADO (dni, nombre, tlf, …, laboratorio, despacho)
CP: dni
VNN: nombre, tlf

CATEGORIA (dniEmpleado, tipo)
CP: (dniEmpleado, tipo)
CAj: dniEmpleado → EMPLEADO(dni)

B. Generar una tabla para el supertipo y otra por cada subtipo, con sus atributos correspondientes

- Esta solución la adoptaremos cuando existen muchos atributos distintos entre los subtipos y, aun así, se quieren mantener los atributos comunes en una tabla

Ejemplo 1



DOCUMENTO (idDocumento, título, idioma, …, categoría)
CP: idDocumento
VNN: categoría

ARTICULO (idArtículo, tema, nPaginas, …)
CP: idArtículo
CAj: idArtículo → DOCUMENTO (idDocumento)

LIBRO (idLibro, editorial, encuadernación, …)
CP: idLibro
CAj: idLibro → DOCUMENTO (idDocumento)

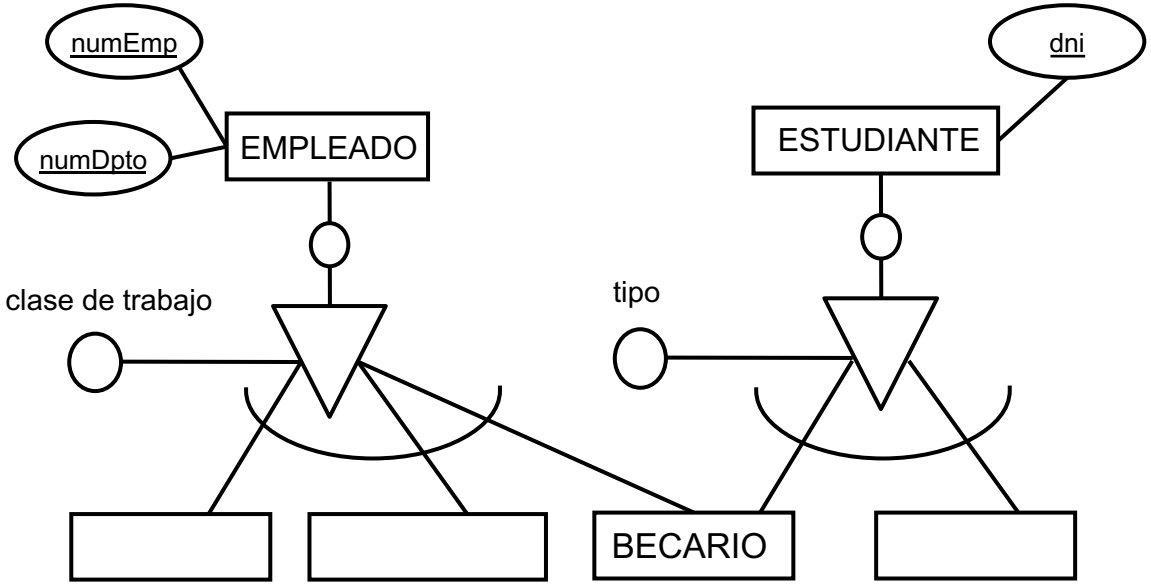
REVISTA (idRevista, volumen, número, …)
CP: idRevista
CAj: idRevista → DOCUMENTO (idDocumento)

- Se puede omitir el atributo discriminante en la tabla del supertipo (ya que puede producir inconsistencias). No obstante, hay que hacer uso de disparadores para mantener la consistencia de la base de datos

- Esta opción, con las variantes adecuadas, se puede aplicar en todos los caso:s especializaciones totales o parciales y especializaciones con o sin solapamiento

- Esta solución es la que más semántica recoge

Ejemplo 2



EMPLEADO (numEmp, numDpto, nombre, …)
CP: (numEmp, numDpto)

ESTUDIANTE (dni, nombre, …)
CP: dni

Alternativa

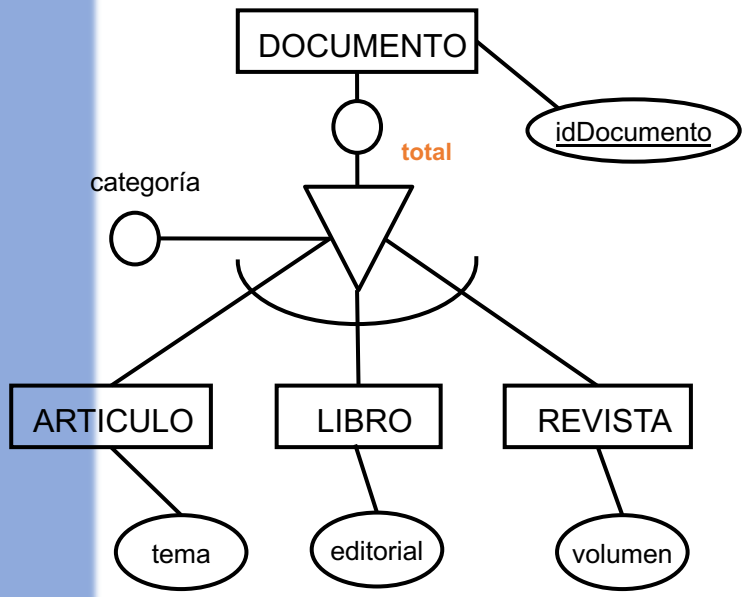
CP: (numEmp, numDpto)
Único: dni
VNN: dni

BECARIO (numEmp, numDpto, dni, fechaInicio, fechaFin, …)
CP: dni
Único: (numEmp, numDpto)
VNN: numEmp, numDpto
CAj: dni → ESTUDIANTE
(numEmp, numDpto) → EMPLEADO

C. Considerar tablas distintas para cada subtipo que contengan, además, los atributos comunes

- Se elegirá esta opción cuando se produzca la misma situación que anteriormente y los accesos que se van a realizar sobre los distintos subtipos siempre afectan a atributos comunes

Ejemplo



ARTICULO (idArticulo, titulo, idioma, …, tema, nPaginas, …)
CP: idArticulo

LIBRO (idLibro, titulo, idioma, …, editorial, encuadernacion, …)
CP: idLibro

REVISTA (idRevista, titulo, idioma, … volumen, numero, …)
CP: idRevista

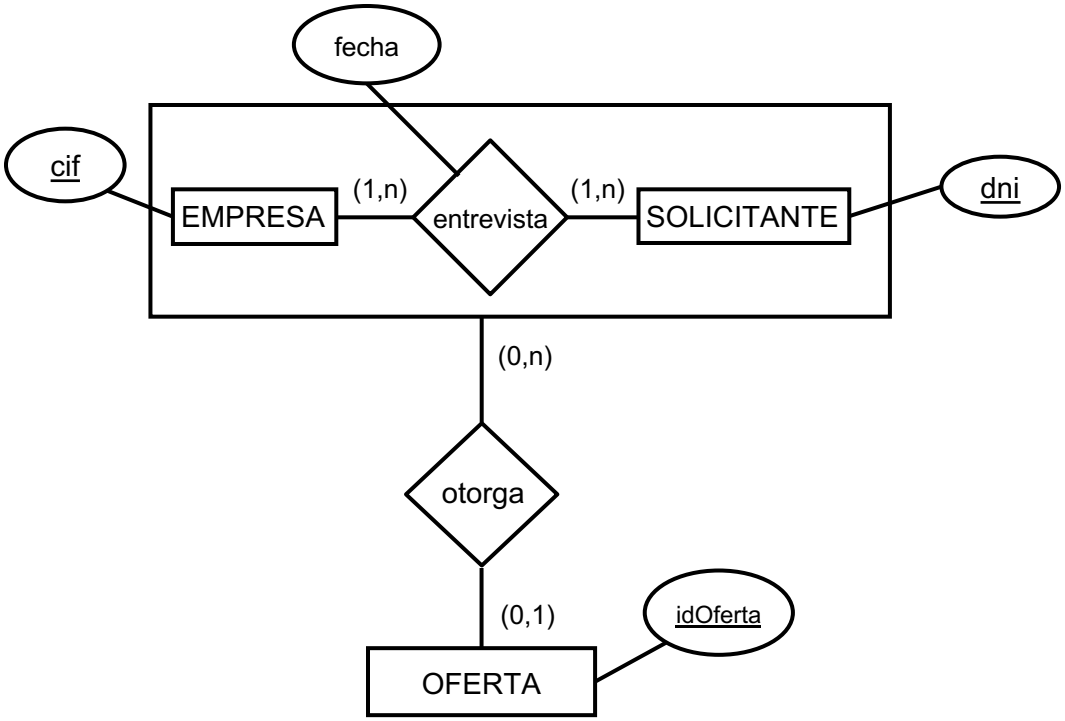
- Esta opción es válida para especializaciones totales con o sin solapamiento. En el caso de solapamiento se introduce mucha redundancia que debe ser controlada si queremos evitar inconsistencias
- En caso de especialización parcial, esta solución es incorrecta porque sería imposible almacenar las ocurrencias del supertipo no especializadas

- La transformación de la especialización conlleva un conjunto de reglas de inserción y borrado
  - La inserción de una ocurrencia en un supertipo implica que se ha de insertar, de forma automática, en todos los subtipos para los cuales dicha ocurrencia satisface la condición por la que se produce la especialización
  - La inserción de una ocurrencia en un supertipo de una especialización total implica que la ocurrencia se insertará obligatoriamente en alguno de los subtipos de la especialización. Si la especialización es sin solapamiento, se ha de insertar únicamente en uno de los subtipos
  - El borrado de una ocurrencia de un supertipo implica que, automáticamente, se elimina de los subtipos a los que pertenece
  - El borrado de una ocurrencia de un subtipo implica borrar la ocurrencia correspondiente del supertipo si la especialización es total (con o sin solapamiento) y es el último subtipo que queda del supertipo correspondiente

2.5.2. Agregación

- Esta transformación es inmediata puesto que se trata de una relación entre entidades. Se aplicarán las reglas de transformación de relaciones

Ejemplo



EMPRESA (cif, nombre, sector, …)
CP: cif

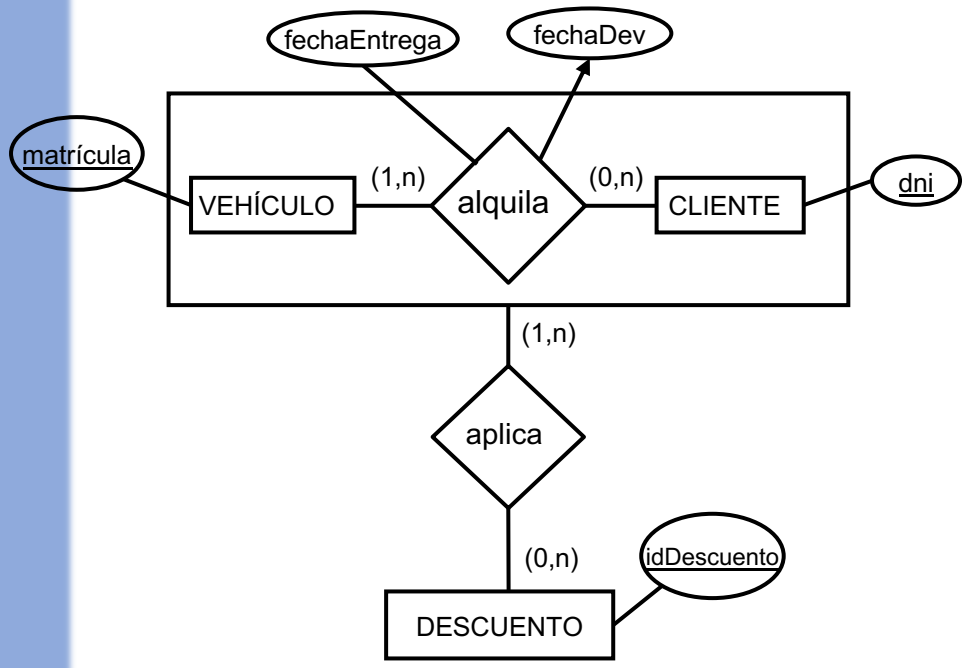
SOLICITANTE (dni, nombre, fechaNac, …)
CP: dni

ENTREVISTA (cif, dni, fecha, oferta)
CP: (cif, dni)
VNN: fecha
CAj: cif → EMPRESA
dni → SOLICITANTE
oferta → OFERTA (idOferta)

OFERTA (idOferta, salario, ...)
CP: idOferta

Suponemos que una misma oferta de trabajo puede necesitar más de un solicitante

Ejemplo



La cardinalidad (0,n) del lado del cliente representa que se desea mantener un histórico de los alquileres

VEHÍCULO (matrícula, marca, modelo, …)
CP: matrícula

CLIENTE (dni, nombre, fechaNac, …)
CP: dni

ALQUILA (matrícula, dni, fechaEntrega, fechaDev)
CP: (matrícula, fechaEntrega)
VNN: dni
CAj: matrícula → VEHÍCULO
dni → CLIENTE

DESCUENTO (idDescuento, porcentaje, ...)
CP: idDescuento

APLICA (matrícula, fechaEntrega, idDescuento, ...)
CP: (matrícula, fechaEntrega, idDescuento)
CAj: (matrícula, fechaEntrega) → ALQUILA
idDescuento → DESCUENTO

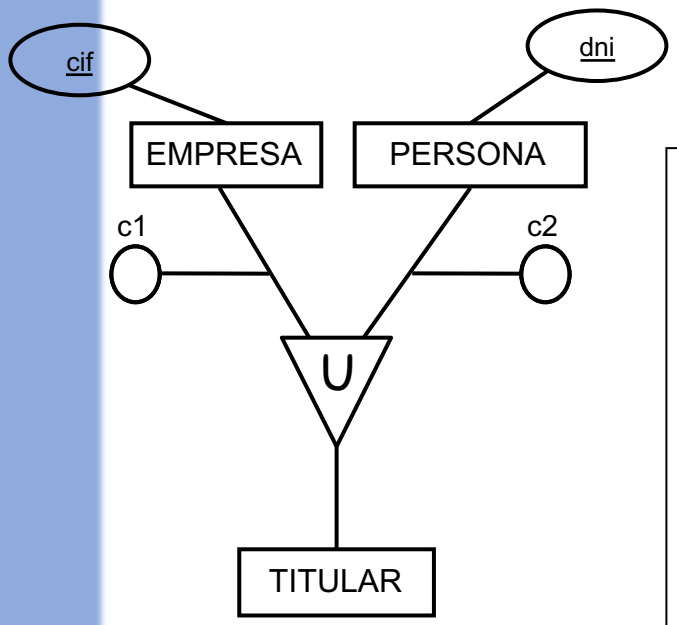
2.5.3. Categoría

- En una categoría, las superclases suelen tener distintos tipos de claves primarias, ya que representan conceptos diferentes

- Si tienen claves distintas, se realizan las siguientes acciones:
  - Especificar un nuevo atributo clave primaria (clave sustituta) para la tabla de la categoría. En esa tabla se incluirán los atributos propios de la categoría
  - Incluir la clave sustituta como clave ajena de las superclases para establecer la relación entre la clave sustituta y la clave de cada superclase
  - En las categorías parciales hay que comprobar que, si la superclase cumple la condición, la clave sustituta no puede almacenar un nulo en las tablas de las superclases

- En una categoría cuyas superclases tengan la misma clave no se necesita de clave sustituta

Ejemplo



Si la categoría fuese total, habría que añadir restricciones VNN en los atributos idTitular de las tablas EMPRESA y PERSONA

EMPRESA (cif, nombre, dirección, …, idTitular)
CP: cif
CAj: idTitular → TITULAR
Único: idTitular
RESTRICCIÓN empresaTitular
  CHECK ((C1 AND idTitular IS NOT NULL) OR (NOT C1 AND idTitular IS NULL))

PERSONA (dni, nombre, estadoCivil, …, idTitular)
CP: dni
CAj: idTitular → TITULAR
Único: idTitular
RESTRICCIÓN personaTitular
  CHECK ((C2 AND idTitular IS NOT NULL) OR (NOT C2 AND idTitular IS NULL))

TITULAR (idTitular, …)
CP: idTitular

Es necesario diseñar e implementar algún tipo de control (por ejemplo, un disparador) para no permitir el mismo valor del campo "idTitular" en las tablas EMPRESA y PERSONA

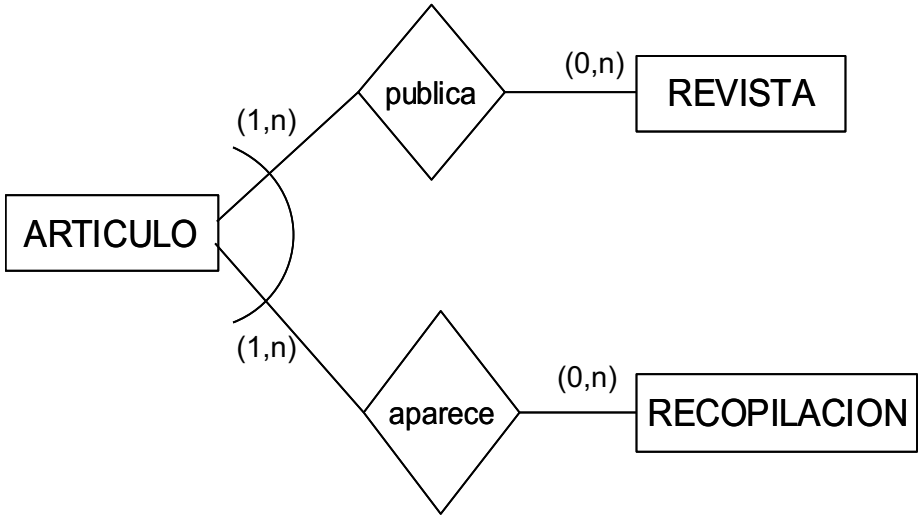
2.5.4. Relaciones exclusivas

- Se gestionan mediante algún mecanismo del SGBD

Ejemplo

PUBLICA (idArtículo, idRevista, …)
CP: (idArtículo, idRevista)
CAj: idArtículo → ARTÍCULO
idRevista → REVISTA

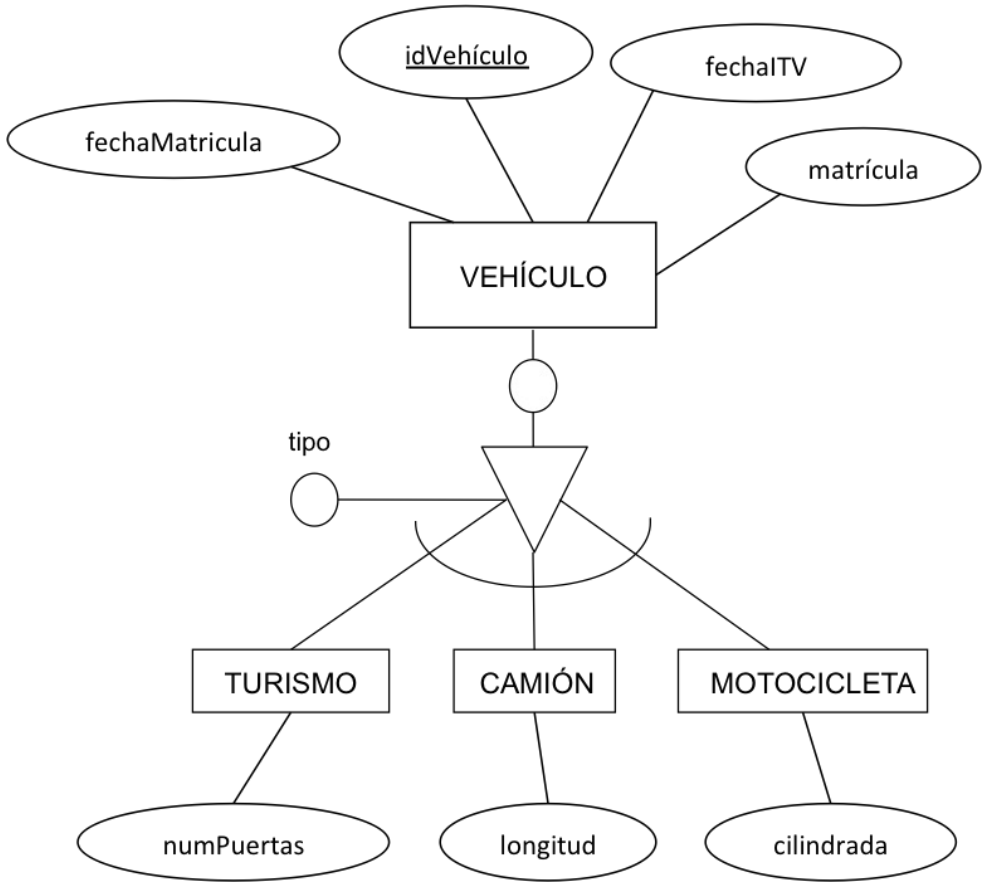
APARECE (idArtículo, idRecopilación, …)
CP: (idArtículo, idRecopilación)
CAj: idArtículo → ARTÍCULO
idRecopilación → RECOPLIACIÓN



Es necesario diseñar e implementar algún tipo de control (por ejemplo, un disparador) para no permitir el mismo valor del campo "idArtículo" en las tablas PUBLICA y APARECE

Ejercicio 1

Realizar la transformación del siguiente fragmento de un esquema conceptual al modelo lógico relacional estándar, utilizando la estrategia de crear una única tabla con el "supertipo" incluyendo el atributo "tipo"



Ejercicio 2

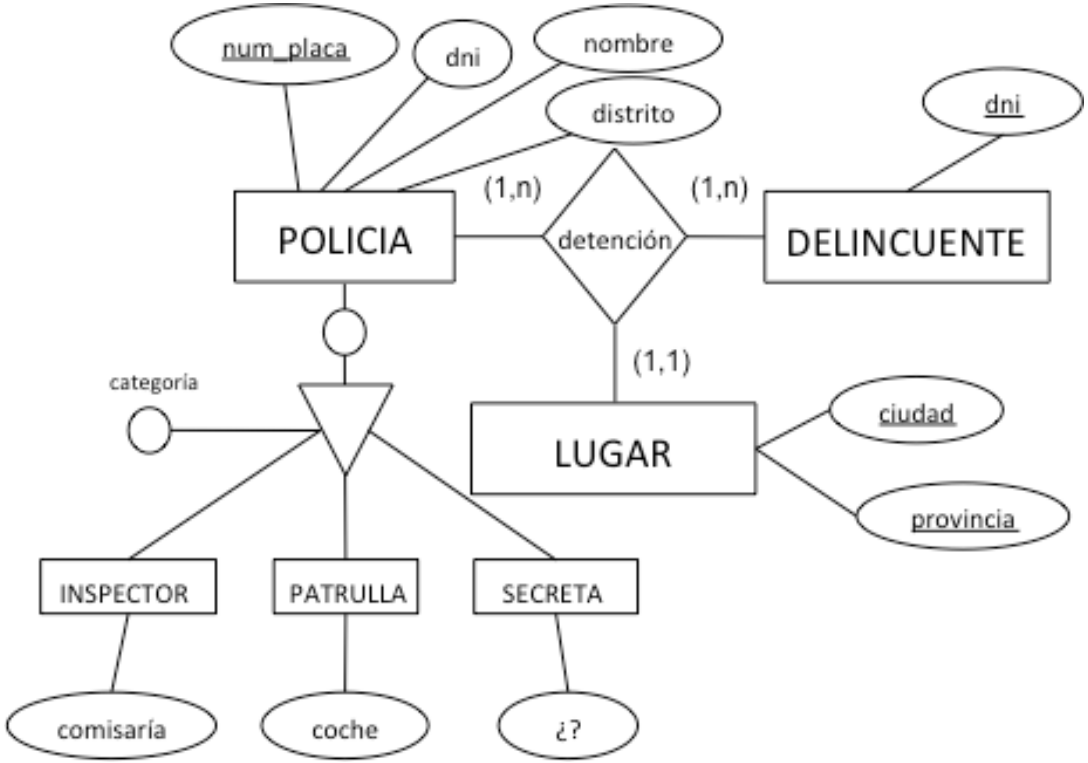
Realizar la transformación del siguiente fragmento de esquema conceptual al modelo lógico relacional estándar

NOTA. Para mantener la claridad del esquema, sólo se han representado algunos de los atributos de las entidades. Los distintos tipos de policías sólo difieren en el atributo que tienen representado.

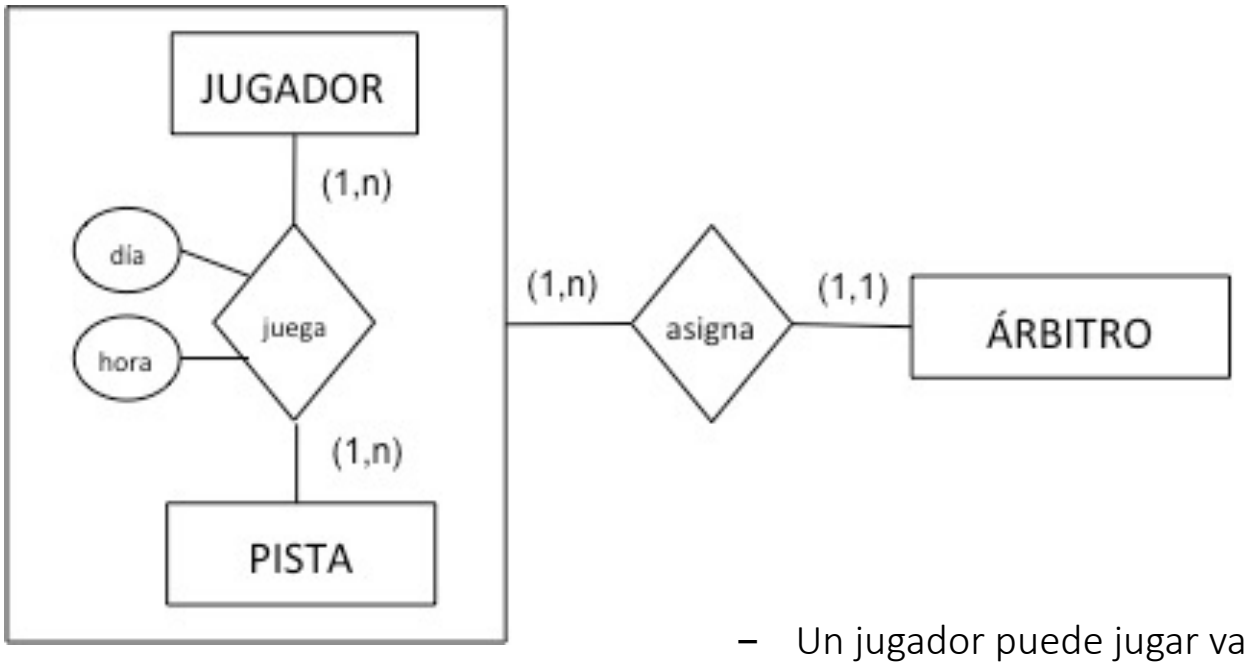
Una vez realizada la transformación, responder, de forma razonada, a las siguientes cuestiones:

a) ¿Un policía puede detener al mismo delincuente más de una vez?. En caso negativo, proponer una solución en el esquema conceptual y realizar su correspondiente transformación al modelo lógico

b) ¿Sería posible que la entidad LUGAR tuviera cardinalidad (0,1), es decir, que no fuera obligatorio almacenar el lugar de la detención?



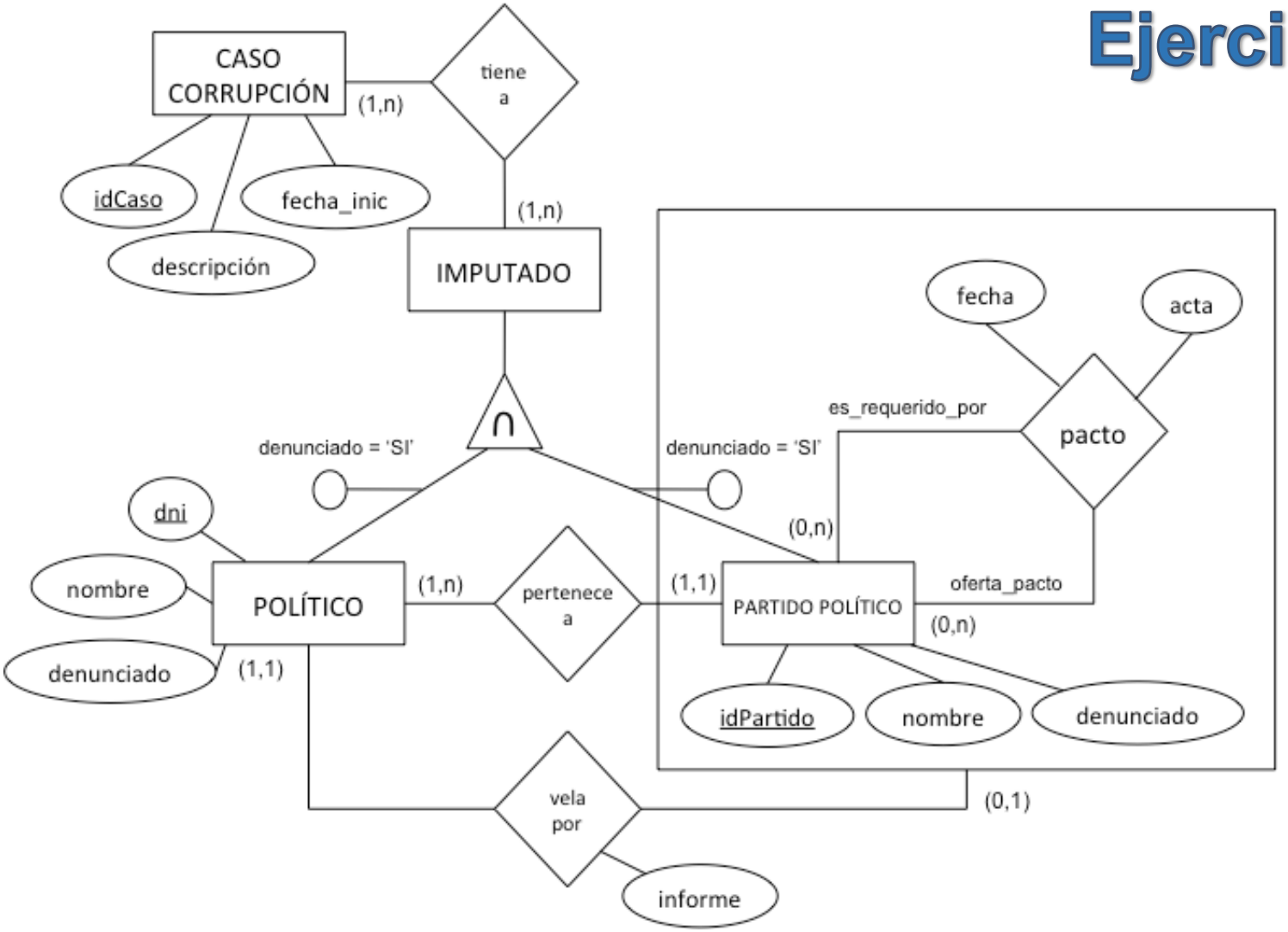
Ejercicio 3



- Un jugador puede jugar varios partidos pero siempre en días distintos
- En un mismo partido juegan hasta 4 jugadores a la vez
- Un árbitro no puede arbitrar más de un partido a la vez

- Cuando se establece un pacto, a éste se le asigna un político que velará para que se cumpla lo pactado
- Los campos "informe" y "acta", de las relaciones "vela por" y "pacto" son atributos simples que representan un fichero pdf
- Ni todos los políticos ni todos los partidos políticos están imputados (menos mal)

Ejercicio 4



2.6. Consideraciones de Diseño

- Una vez obtenido el esquema Entidad-Relación (E-R) de la "realidad" que se desea modelar, es conveniente revisarlo detalladamente con el fin de refinarlo para que el esquema lógico que se obtenga sea lo más eficiente posible

- Para conseguir esto, se tratarán algunas cuestiones significativas que pueden producirse durante la realización del diseño conceptual

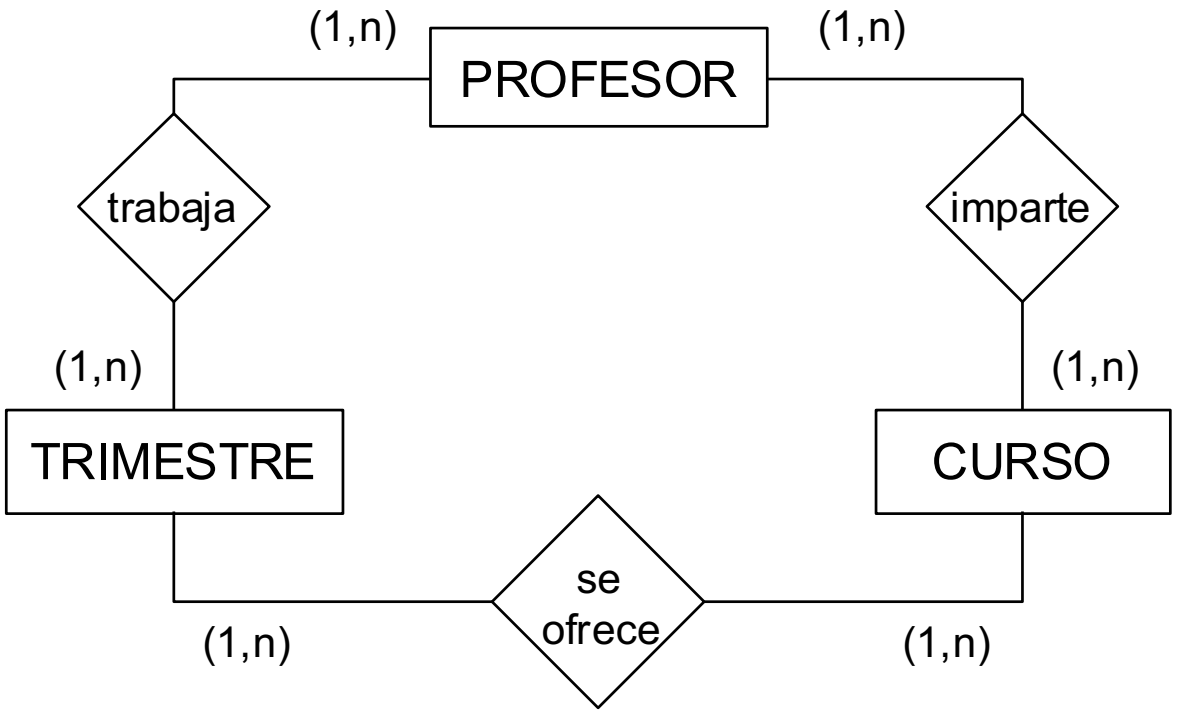
2.6.1. Relaciones de grado mayor que dos

- En ocasiones, al representar la información mediante el modelo E-R, podemos encontrarnos con situaciones en las que algunas de las relaciones que se establecen entre las entidades no recogen todo el contenido semántico que se pretende

- **Ejemplo**: se desea modelar la información sobre una academia donde los profesores imparten cursos repartidos en trimestres, con las siguientes restricciones:

  - Un profesor imparte, en general, varios cursos y un curso puede ser impartido por varios profesores
  - Los cursos pueden impartirse en diferentes trimestres realizándose, en cada trimestre, varios cursos
  - Se desea también conocer en qué trimestre está trabajando un profesor, teniendo en cuenta que un profesor puede impartir cursos en diferentes trimestres y trabaja todos los trimestres
  - En general, se desea saber la información de los profesores con respecto a los cursos que imparten en cada trimestre

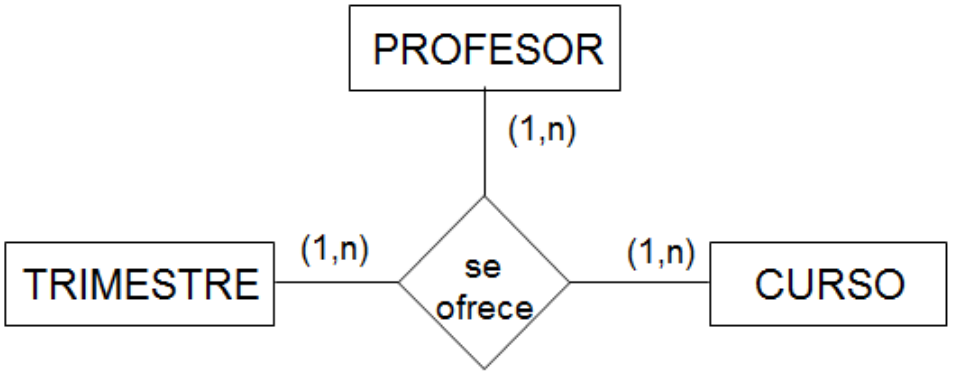
- Si se modela esta información tal y como se va leyendo, lo más lógico es que surja el siguiente diagrama E-R:



- Sin embargo, mediante esta representación no se obtiene la información de un profesor respecto a un curso y a un trimestre

- Por ejemplo, el hecho de tener las tuplas (p1, c3), (p1, t2) y (c3, t2) no implica que exista la información (p1, c3, t2), ya que el profesor **p1** ha podido impartir el curso **c3** en otro trimestre y ha trabajado en el trimestre **t2** impartiendo un curso distinto al **c3**

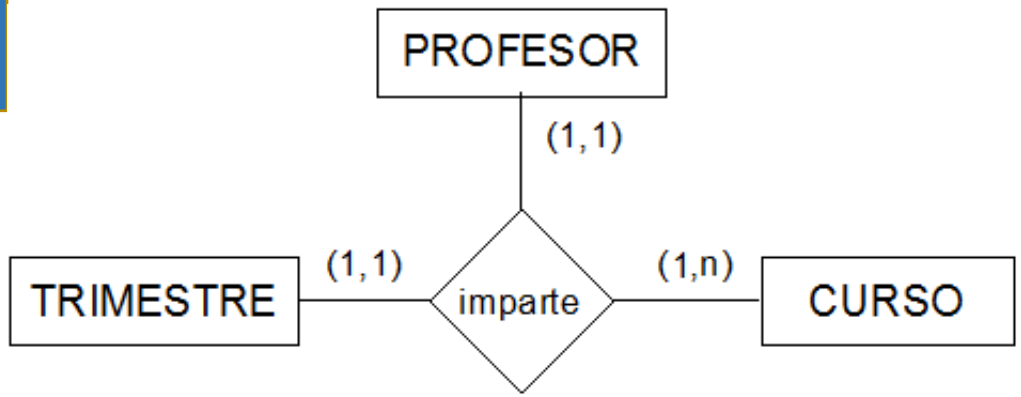
El diseño anterior no es correcto, ya que no permite recoger toda la semántica del problema. En este caso se debería modelar con una relación ternaria



- Veamos otro ejemplo en el que se quiere modelar la información sobre la misma academia. Se desea obtener la información de los cursos que un determinado profesor imparte durante un trimestre, pero ahora con las siguientes restricciones:

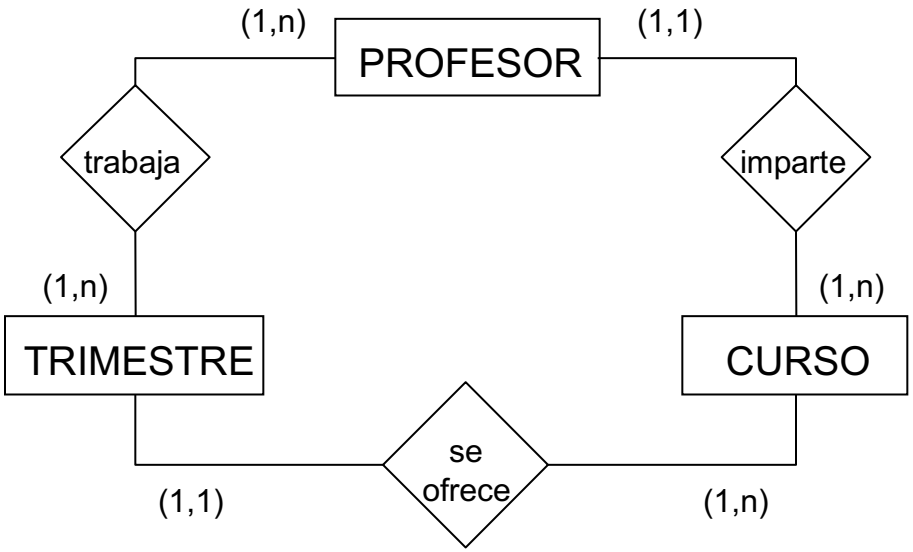
  - Un curso solo lo imparte un profesor y solo se imparte en un trimestre
  - Un profesor en un trimestre puede impartir varios cursos

Supongamos que, este caso, la primera opción elegida es construir una relación ternaria



- Esta relación ternaria, al contrario que la anterior, sí se puede descomponer en tres relaciones binarias sin perder contenido semántico

- Esta descomposición debe realizarse (cuando sea posible) ya que, cuantas menos entidades participen en una relación, más eficiente será controlar la integridad referencial de los datos y, además, en algunos casos será posible eliminar algunas relaciones del esquema sin pérdida de semántica



2.6.2. Control de Redundancias

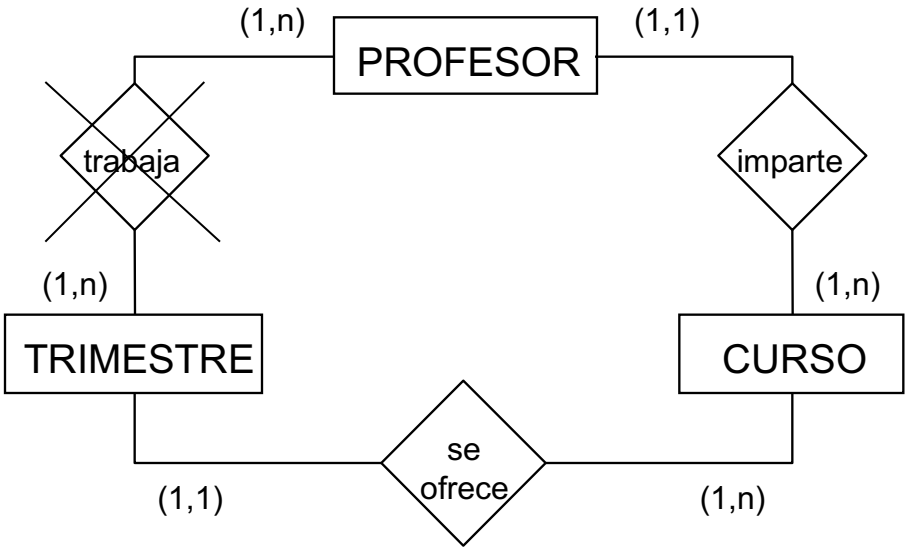
- Una vez realizado el diagrama conceptual o esquema E-R, hay que analizar si presenta redundancias, ya que éstas pueden generar situaciones erróneas o anómalas en la Base de Datos

- En un esquema E-R hay que tener especial precaución en los siguientes puntos:

  - Existencia de atributos redundantes que se derivan de otros mediante algún cálculo:
    - Deben tenerse en cuenta y marcarlos como atributos derivados

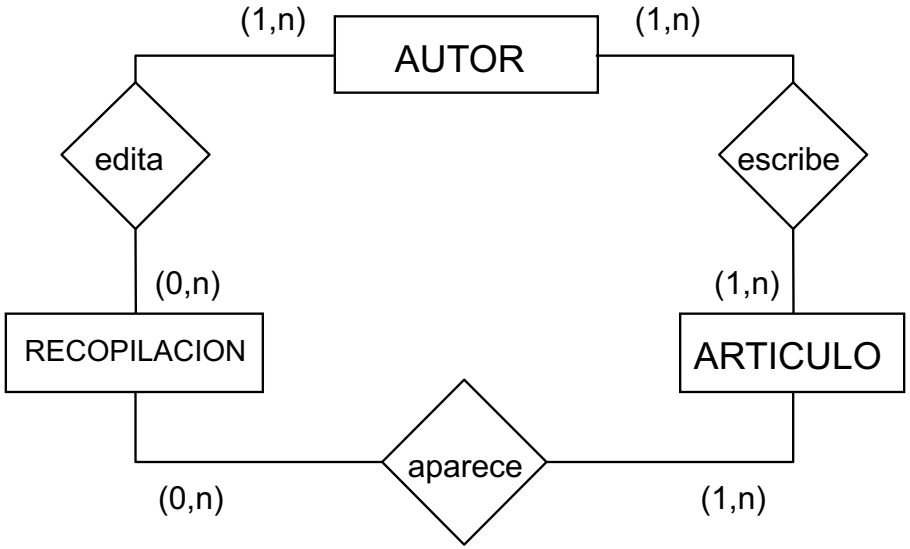
  - Existencia de ciclos:
    - Los ciclos pueden indicar la existencia de relaciones redundantes

- En el esquema anterior se produce un ciclo entre las entidades PROFESOR, TRIMESTRE y CURSO, por lo que pueden existir relaciones redundantes



- Si se conocen los cursos que imparte un profesor y el trimestre en los que se imparten, se puede deducir en qué trimestre ha trabajado dicho profesor
- Dado un trimestre, si se conocen los cursos que se han impartido en éste, podemos deducir qué profesores han trabajado en dicho trimestre
- Por lo tanto, la relación TRABAJA, entre las entidades PROFESOR y TRIMESTRE, es redundante

- Sin embargo, puede ocurrir que, a pesar de existir un ciclo, no existan relaciones redundantes



- Realizando un análisis similar al efectuado en el diagrama anterior, se puede comprobar que no se puede eliminar ninguna de las relaciones sin pérdida de semántica

- Se comprueba que se trata de un ciclo no redundante

2.6.3. Dimensión Temporal

- El tratamiento de la dimensión temporal en bases de datos es un tema complejo sobre el que existen diversas propuestas

- Es indudable la necesidad de establecer un método gráfico que recoja, en el esquema conceptual, el transcurso del tiempo y su influencia en la variación de los datos

- Al realizar el esquema conceptual se debe analizar si los datos que se pretenden almacenar van a constituir una base de datos histórica o, si por el contrario, sólo interesa el estado actual de los datos

- El esquema de la figura 1.a representa un modelo en el que sólo interesa el estado actual de cada ejemplar:

  - Si está o no alquilado y quién fue el último socio que lo alquiló

- El esquema de la figura 1.b representa un modelo en el que interesa mantener un histórico de quién ha pedido prestado cada ejemplar en cada momento, así como las fechas de préstamo y devolución

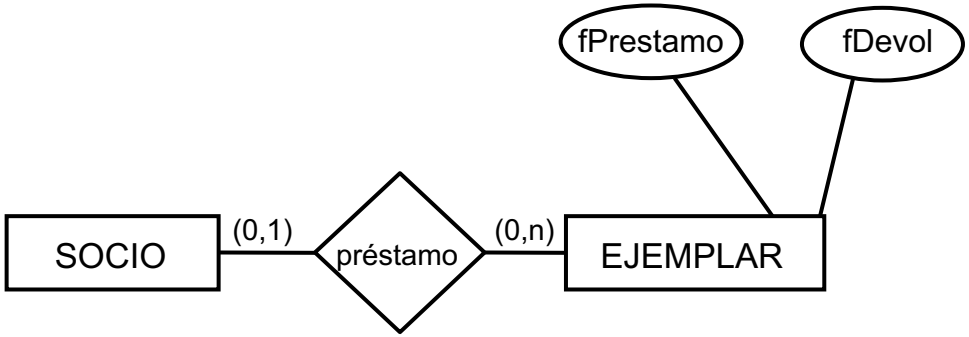


Figura 1.a

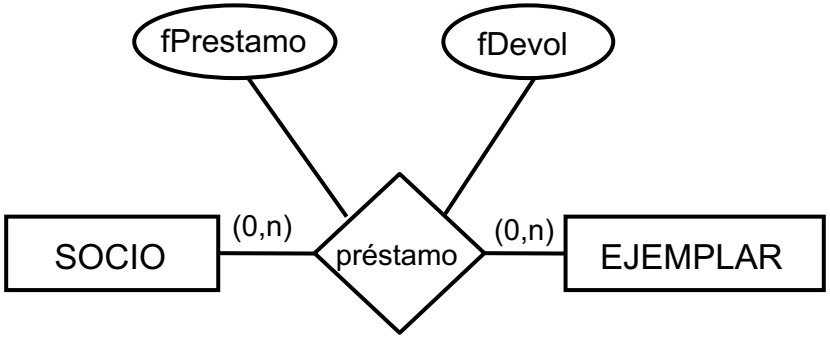


Figura 1.b

Ejercicio 1 - Boletín

La galería de arte ART-ATTACK desea informatizar la gestión de sus recursos. El principal activo de la empresa son las obras de arte. De ellas se desea mantener un código de obra, un nombre y una fecha de adquisición. Generalmente, las obras de arte son pinturas y esculturas, aunque hay otras obras que no se pueden considerar encuadradas en estas categorías. De las pinturas guardaremos el estilo y la fecha de creación, mientras que de las esculturas se desea almacenar el material de construcción. Además, si conocemos el autor de la obra también almacenaremos su información personal.

La galería ofrece sus obras para realizar exposiciones. Habrá que saber la fecha de comienzo y final de cada exposición y las ciudades de celebración (teniendo en cuenta que existen exposiciones itinerantes, es decir, que se celebran en diferentes ciudades). La galería puede ceder una obra de arte a distintas exposiciones (evidentemente en fechas distintas), por lo que es necesario saber en qué exposiciones ha estado una obra de arte. Una exposición sólo se compone de pinturas o esculturas, pero no hay exposiciones mixtas.

Por otra parte, la galería también puede vender sus obras. Los compradores pueden ser personas individuales o instituciones (Ministerio de Cultura, Ayuntamientos, etc.). De las personas se almacena el nombre, D.N.I. y dirección. De las instituciones deseamos conocer el nombre y el C.I.F. Cada vez que se realiza la venta de una obra de arte a un comprador se levanta un acta notarial para certificar, entre otras cosas, la autenticidad de la venta. La galería desea saber quién es el notario que firma cada acta por lo que trabaja con un grupo de notarios de los que se almacena su número de colegiado, dirección y teléfono. En principio, las obras de arte no tienen precio. Cuando se realiza la venta se almacena la fecha de venta y el precio acordado con el comprador.

Ejercicio 17 - Boletín

Una empresa de comidas a domicilio necesita crear una base de datos para mantener toda la información que genera, teniendo en cuenta las siguientes restricciones y reglas.

En la empresa se distinguen varios tipos de empleados. De todos ellos es necesario almacenar su dni, nombre, apellidos y teléfono móvil. Además, de los cocineros se almacenará también su fecha de alta en la empresa y de los pinches interesa conocer su titulación. De los repartidores habrá que saber qué tipo de carné de conducir poseen. Del resto de trabajadores solo almacenaremos la información general. Hay que tener en cuenta que todos los pinches están al cargo de un cocinero.

De los platos que elabora la empresa hay que almacenar su nombre, su precio y los ingredientes con los que se elabora junto con su medida (1/2 kg., dos cucharadas, 3 porciones, un manojo, etc.). La empresa dispone de varios almacenes en los que almacena los ingredientes. Cada uno de ellos tiene un nombre, una capacidad y es de un determinado tipo (despensa, cámara frigorífica, etc.). Además, cada almacén tiene estanterías que se identifican con una letra y un dígito ('A3', 'C1', 'E7', etc.) y de ellas guardamos, también, sus longitudes. Evidentemente, distintos almacenes pueden tener estanterías con las mismas letras. Hay que saber en qué estanterías están los ingredientes y cuánta cantidad hay en cada una de ellas (10 litros de leche, 4 paquetes de azúcar, etc.). En cuanto a la elaboración, necesitamos saber qué cocineros saben cocinar cada plato.

Respecto a los pedidos, queremos ir almacenando la información personal de los clientes y la información de los pedidos que se van realizando (identificador de pedido, fecha, hora, etc.). Los pedidos están formados por platos, así que habrá que saber cuáles son y qué cantidad hay de cada uno de ellos. También hay que saber el repartidor que lo llevará al domicilio del cliente. Dentro de un pedido, cada plato puede tener (o no) un descuento en función de ciertas variables (día de la semana, cantidad de platos pedidos, etc.). Los descuentos están tipificados en la base de datos, almacenándose su descripción, porcentaje de descuento, etc. Hay que saber qué descuento se le ha aplicado a cada plato dentro de un pedido.

Ejercicio 20 – Boletín. La empresa de buceo "ETSIDiving" desea diseñar una base de datos para gestionar su información, con las siguientes restricciones y reglas que cumplir.

Las personas implicadas en la empresa son los buzos, que pueden ser profesionales o aficionados. De todos ellos se almacenarán sus datos personales (dni, nombre, apellido, teléfono de contacto, …), sus títulos (open water, 1 estrella, dos estrellas, etc.) y las fechas en la que obtuvieron cada uno de ellos. La empresa tiene establecido los lugares en las que se realizan las inmersiones. Estos lugares están tipificados como deportivos o de trabajo. De todos ellos habrá que almacenar un código que lo identifique y su nombre (por ejemplo, isla de Bacuta, playa de Mazagón, etc.).

La empresa desea tener almacenada la información de las inmersiones que realizan los buzos en los distintos lugares, teniendo en cuenta que los buzos aficionados solo realizan inmersiones en los lugares tipificados como deportivos y los profesionales solo lo hacen en los de trabajo. Tanto en una como en otra, es necesario almacenar la fecha en la que se ha realizado la inmersión. En una inmersión deportiva tienen que ir entre 3 y 5 buzos, mientras que las inmersiones de trabajo la hacen siempre 2 parejas de buzos profesionales ya que, por normativa, éstos siempre deben bucear en de dos en dos. La empresa necesita saber cómo están formadas estas parejas, teniendo en cuenta que son fijas y que un buzo solo puede pertenecer a una pareja.

Cada vez que un buzo aficionado realiza una inmersión a un determinado lugar, la empresa le puede ceder en préstamo algún material, pero no es obligatorio puesto que cada buzo puede utilizar el suyo propio. Es necesario saber en qué inmersión se ha realizado el préstamo al buzo. En caso de que se le preste, hay que almacenar la fecha de entrega y devolución. El material de la empresa está almacenado en la base de datos de forma individual, es decir, cada elemento tiene su propio código (un neopreno con código M123, una máscara con código M432, unas aletas con código M777, etc.). Cuando un material se estropea, se lleva a una empresa de reparación, así que hay que saber en qué empresas está (o ha estado) cada material junto con su fecha de entrada y fecha de devolución una vez reparado.