

# Bases de Datos y Sistemas de Información

Grado en Ingeniería Informática

Unidad Didáctica 4: Diseño de Bases de Datos  
Relacionales

Parte 2: Diseño Conceptual

(Doc. UD4.2)

Curso 2021/2022



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Para la elaboración de este documento se han consultado los siguientes textos:

- [BRJ06] Booch, G.; Rumbaugh, J.; Jacobson, I.; El lenguaje unificado de modelado (2ª edición). Pearson, Addison Wesley, 2006.
- [CB10] Thomas M. Connolly, Carolyn E. Begg; Database Systems: A Practical Approach to Design, Implementation and Management, 5/E , Addison-Wesley, 2010.
- [CCM03] Celma, M.; Casamayor, J. C.; Mota, L.; Bases de datos relacionales. Pearson, Prentice Hall, 2003.
- [EN02] Ramez Elmasri, Shamkant B. Navathe. Fundamentos de sistemas de bases de datos. Addison-Wesley 2002)
- [Lar04] Larman, C.; UML y Patrones: una introducción al análisis y diseño orientado a objetos y al proceso unificado (2ª edición). Pearson, Prentice Hall, 2004
- [MCC94] Mota, L.; Celma, M.; Casamayor, J. C.; Bases de datos relacionales: teoría y diseño SPUPV 767.94, 1994.
- [Oli01] Olivé, A.; Modelització conceptual de sistemes d'informació. Edicions UPC. 2001.
- [SR07] Stevens, P.; Pooley, R.; Utilización del UML en Ingeniería del Software con objetos y componentes (2ª edición). Pearson, Addison Wesley, 2007.

## Índice

1	Introducción.....	1
2	Diseño Conceptual.....	1
2.1	El diagrama de clases de UML.....	1
2.2	Obtención del diagrama de clases .....	9
3	Ejercicios resueltos .....	16
3.1	Restaurante.....	16
3.2	Ciclismo .....	17
4	Ejercicios para resolver .....	18
4.1	Una compañía de seguros.....	18
4.2	Una galería de pintura .....	18
4.3	Biblioteca .....	18
4.4	La revista “La Periódica” .....	19
4.5	Una compañía de alquiler de vehículos (Prácticas).....	19
4.6	Un supermercado (Prácticas).....	20
4.7	El club de vela de la UPV .....	21
4.8	Una escuela de enseñanza obligatoria.....	21
4.9	Parque-Acuario .....	22
4.10	Una empresa de construcción .....	22
4.11	Una empresa de desguace .....	23
4.12	Control de llamadas (Prácticas).....	23
4.13	La compañía “SoftWWW” (Prácticas) .....	24
4.14	Una compañía de líneas aéreas .....	24
4.15	Control de finanzas .....	25
4.16	Centro de acogida de perros.....	25
4.17	Una compañía eléctrica .....	26
4.18	Una cooperativa de vinos.....	26
4.19	Elecciones a rector .....	26
4.20	Camino de Santiago .....	27
4.21	Locales de restauración .....	27
4.22	Conflictos bélicos .....	28

## 1 INTRODUCCIÓN

En este documento se presenta el Diseño conceptual de Sistemas de Información. Para realizar esta tarea se necesita una herramienta que permita obtener una representación (usualmente gráfica) de la información que se maneja en el sistema de información. La herramienta que se propone utilizar es el UML (*Unified Modeling Language*)<sup>1</sup>. El UML fue propuesto por Booch, Rumbaugh y Jacobson en la década de los 90 y pretende ser una herramienta para modelar desde sistemas de información empresariales hasta aplicaciones distribuidas basada en web. El UML ya ha sido introducido en la asignatura de Ingeniería del Software (semestre 3A), pero de todos los diagramas que permite, sólo se va a utilizar el *diagrama de clases* para el diseño de la parte estática (los datos con sus restricciones) de un SI.

## 2 DISEÑO CONCEPTUAL

El diseño conceptual es la fase del diseño de una base de datos cuyo objetivo es *“obtener una representación de la realidad que capture las propiedades estáticas y dinámicas de la misma que son necesarias para satisfacer los requisitos; esta representación debe suponer una imagen fiel del comportamiento del mundo real”*.

De entre todos los diagramas que permite generar el UML hay uno, el *diagrama de clases*, que resulta muy útil en el diseño de una base de datos para la representación del esquema conceptual.

### 2.1 El diagrama de clases de UML

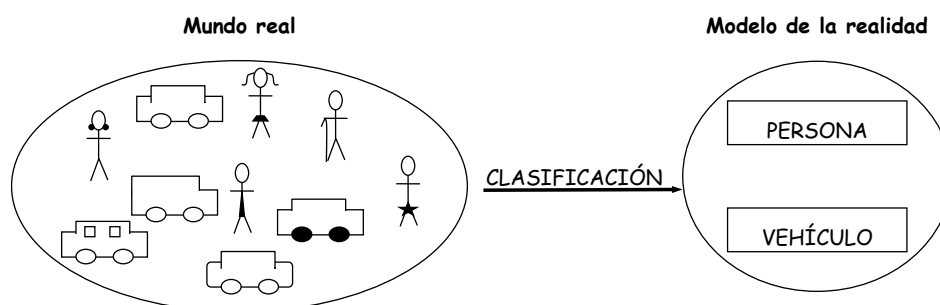
El diagrama de clases permite representar las estructuras que constituyen el contenido del sistema de información junto con restricciones de distintos tipos que limitan las ocurrencias válidas de las mismas. Para ello hace uso, fundamentalmente, de tres conceptos: clase, atributo y asociación. Además, para aumentar la capacidad expresiva del diagrama también se contempla la definición de objetos especializados (o generalizados). Todos estos conceptos se presentan a continuación con detalle.

#### 2.1.1 Clase

La observación de la realidad permite detectar el conjunto de “objetos” (físicos o conceptuales) de los que se quiere almacenar información para, mediante el uso de la clasificación, que es uno de los mecanismos de abstracción más primarios que existen, descubrir el conjunto de “clases” (o tipos de objetos) que son de interés para la organización. Este mecanismo de abstracción, que es utilizado la mayor parte de las veces de forma inconsciente, permite no prestar atención a las ocurrencias concretas sino al conjunto de ocurrencias.

##### Ejemplo 1

En el siguiente gráfico se aprecia el uso de este mecanismo de abstracción:



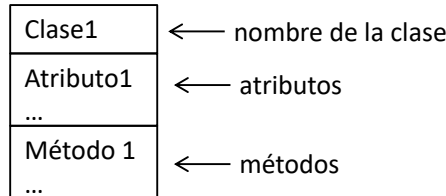
Así pues, los componentes básicos de un sistema de información son los objetos de los que se quiere almacenar información; todos los objetos que son de la misma clase se representan con una clase.

<sup>1</sup> Una alternativa a esta herramienta es el Modelo Entidad-Relación propuesto por Peter Pin-Shan Chen en 1976.

Una clase es la descripción de un grupo de objetos con estructura, comportamiento y relaciones similares.

Con una clase se representará cualquier tipo de persona, concepto, suceso o evento (en definitiva cualquier “cosa”) sobre el que se quiera almacenar información.

En diagrama de clases del UML, una clase se representa con un rectángulo nominado dividido en tres compartimentos como se muestra a continuación.



En el apartado siguiente se describe la sección de los atributos. En la tercera sección de la clase se pueden incluir las operaciones (o procesos) asociadas a la clase. Dado que esto se corresponde con la dinámica del sistema, este punto no se va a presentar y en muchos casos no se dibujará.

### 2.1.2 Atributo

Un atributo representa una propiedad del elemento que se está modelando y que es compartida por todos los objetos de la clase.

Un atributo es una propiedad de una clase identificada con un nombre, cuyas instancias pueden tomar valor de un conjunto que se especifica.

Gráficamente, los atributos se listan en el compartimento que hay justo debajo del nombre de la clase y para cada uno se puede indicar:

- El tipo de datos asociado que determina los valores que puede tomar el atributo. Se pueden dar dos casos:
  - El atributo toma valores simples: en este caso hay que indicar el nombre de un tipo de datos conocido: *char* (cadena de caracteres), *real*, *int* (entero), *date* (fecha), ..., o bien una enumeración de valores posibles entre paréntesis.
  - El atributo está estructurado en un registro.
- Restricciones. Se definen entre llaves y pueden ser:
  - De unicidad: representa el hecho de que las distintas ocurrencias de una clase deben tomar valores distintos para el atributo o conjunto de atributos sobre los que se define esta restricción.  
Se indica con la etiqueta {*único*} en cada atributo que forme parte del iésimo conjunto de atributos con restricción de unicidad
  - De cardinalidad: expresan el número de valores que puede tomar el atributo para cada ocurrencia de la clase. Los valores más usuales son los siguientes:
    - {1..1}: el atributo tiene exactamente un valor para cada ocurrencia de la clase.
    - {0..1}: el atributo puede tener como mucho un valor para cada ocurrencia de la clase o puede no tener valor (es el valor por defecto).
    - {1..\*}: el atributo puede tener más de un valor para cada ocurrencia de la clase pero al menos tiene un valor.
    - {0..\*}: el atributo puede tener más de un valor para cada ocurrencia de la clase o puede no tener valor.
  - De identificación: un identificador es un conjunto de atributos con restricción de unicidad y con cardinalidad {1..1} que permite distinguir entre sí dos ocurrencias cualesquiera de la clase y que se elige como mecanismo para hacer referencia a los

objetos o instancias de la clase (sólo hay un identificador aunque puede constar de varios atributos). A los atributos que constituyen el identificador se les distingue con la etiqueta *{id}*.

## Ejemplo 2

En la figura siguiente se muestra la clase *Persona* que se describe por sus atributos:

- DNI (Documento Nacional de Identidad): se ha elegido como identificador y es de tipo cadena de caracteres.
- NSS (Número de la Seguridad Social): siempre toma un valor que debe ser distinto para cada persona y es de tipo cadena de caracteres.
- Nombre: siempre toma un valor para cada persona y tiene dos campos, *propio* y *apellidos*, ambos de tipo cadena de caracteres.
- Edad: puede tener o no valor para cada persona y es de tipo entero.
- Teléfonos: cada persona puede tener cero, uno o varios números de teléfono.

Persona
DNI: {id}: char NSS: {unico <sub>1</sub> }: {1..1}: char nombre: {1..1}: propio:{1..1}: char apellidos:{1..1}: char edad: {0..1}: int teléfonos: {0..*}: char

### 2.1.3 Asociación

Los objetos de un sistema de información se conectan unos con otros siendo también de interés modelar estas conexiones; para ello se utilizará el concepto de *asociación* del diagrama de clases de UML.

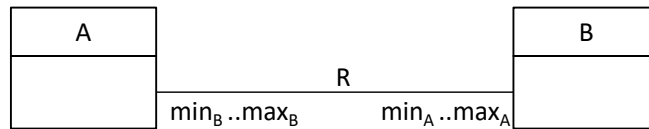
Una asociación es una relación estructural que especifica que los objetos de una clase están conectados con los objetos de otra.

Gráficamente, una asociación se representa con una línea continua que conecta dos clases no necesariamente distintas y puede llevar los siguientes adornos:

- Nombre: una asociación puede tener un nombre que se utiliza para describir la naturaleza de la relación. Se incluye en la parte central de la línea que representa la asociación y, aunque no es obligatorio especificarlo, ayuda a la comprensión del diagrama. Suele ser menos necesario en el caso de que haya roles. Para clarificar la interpretación de la asociación se puede incluir una flecha, ►, que indique el sentido del nombre de la asociación (la navegabilidad). Por simplicidad, y a no ser que quede ambiguo el sentido de la asociación, se omitirá. Sólo se ha incluido en algunos ejemplos para ilustrar su uso.
- Rol: sirve para expresar el papel que juega una clase en una asociación. En el caso de que la asociación de una clase sea consigo misma (*asociación reflexiva*), es necesario incluir el rol que la clase tiene cuando participa en la asociación en cada uno de los dos sentidos posibles ya que, si no se incluye, no se pueden expresar claramente las restricciones de cardinalidad. Para asociaciones entre clases distintas, no es obligatorio incluir esta información.
- Cardinalidad (o multiplicidad): una asociación representa el hecho de que instancias de una clase se conectan, con un determinado significado, con instancias de la otra clase. Es frecuente que existan limitaciones en cuanto al número de conexiones posibles. Estas “limitaciones” se representan en la cardinalidad de cada clase en la asociación y se definen con un valor mínimo y un valor máximo separados por dos puntos seguidos. Cuando se indican estos valores en un extremo de una asociación, se está especificando cuántos objetos de la clase de ese extremo

puede haber para cada objeto de la clase en el otro extremo.

En el siguiente diagrama se han representado dos clases (de nombres *A* y *B*) y una asociación entre ellas (de nombre *R*). Las cardinalidades incluidas especifican lo siguiente:



- Cada ocurrencia de *A* se relaciona con, como mínimo  $min_A$  ocurrencias de *B* y como máximo con  $max_A$  ocurrencias de *B*;
- Cada ocurrencia de *B* se relaciona con, como mínimo  $min_B$  ocurrencias de *A* y como máximo con  $max_B$  ocurrencias de *A*;

Considerando la cardinalidad de *A* en el diagrama anterior, los valores más frecuentes son los siguientes:

- $min_A=1$  y  $max_A=1$ : Cada ocurrencia de *A* se relaciona con, como mínimo una ocurrencia de *B* y como máximo con una ocurrencia de *B*, esto es, cada ocurrencia de *A* se relaciona exactamente con una ocurrencia de *B*.
- $min_A=0$  y  $max_A=^*$ : Cada ocurrencia de *A* puede relacionarse con cualquier número de ocurrencias de *B* (0, 1,...).
- $min_A=0$  y  $max_A=1$ : Cada ocurrencia de *A* se relaciona como máximo con una ocurrencia de *B* o puede no relacionarse con ninguna.
- $min_A=1$  y  $max_A=^*$ : Cada ocurrencia de *A* puede relacionarse con cualquier número de ocurrencias de *B* pero al menos con una (1, 2, ...).

Cuando la cardinalidad mínima de una clase es distinta de 0 se dice que esa clase tiene *restricción de existencia* respecto a la asociación.

En ocasiones para expresar la cardinalidad de una clase en una asociación sólo se proporciona uno de los dos valores, siendo el otro asumido por defecto, así:

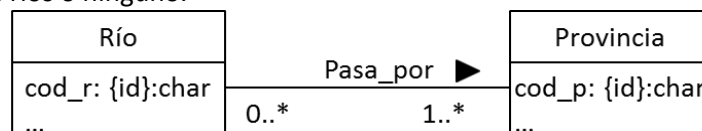
- 1: equivale a 1..1
- \*: equivale a 0..\*

Sin embargo, por claridad es aconsejable especificar los dos valores para cada caso.

Si la relación que representa la asociación tiene también propiedades, éstas se representan en los denominados *atributos de enlace*. Estos atributos se especifican en un recuadro sin nombre que se conecta mediante una línea discontinua con la línea de la asociación.

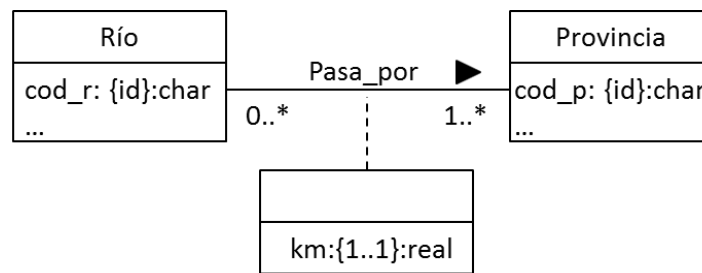
### Ejemplo 3

Sea el siguiente diagrama que representa la información de qué ríos pasan por qué provincias. Se ha representado que un río puede pasar por varias provincias, al menos por una, y que por una provincia puede que pasen varios ríos o ninguno.



Si queremos incluir en el diagrama un atributo que represente la información de “durante cuántos kilómetros pasa cada río por cada provincia que atraviesa”, la asociación tendrá un atributo de enlace:

<sup>2</sup> En UML no se utiliza el \* sino la etiqueta N, con el sentido de muchos. Nosotros hemos conservado el \* para rendir tributo al modelo de datos Entidad Relación que utilizábamos anteriormente.



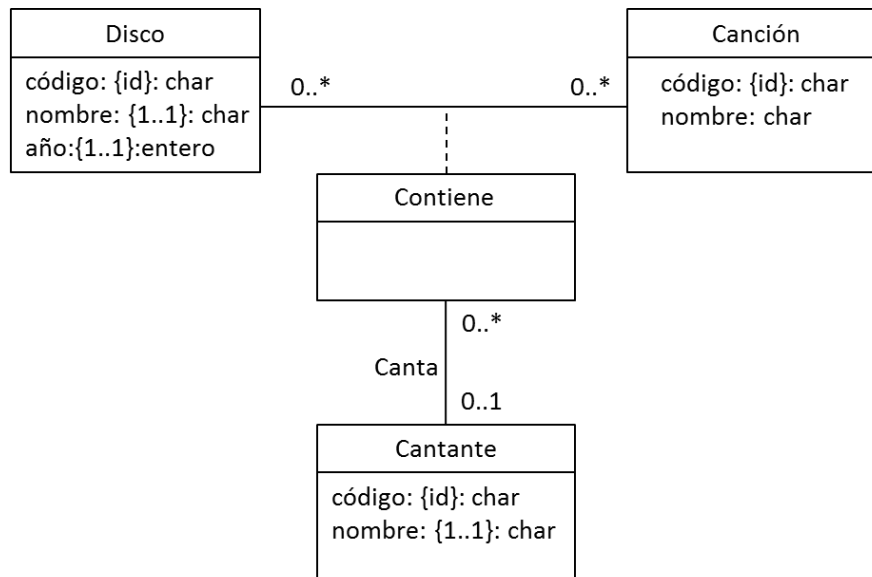
Una asociación puede comportarse como una clase, llamada *clase asociación*, de manera que podrá asociarse con otras clases. Para representar este caso, se incluirá también un recuadro conectado mediante una línea discontinua con la línea de la asociación que puede tener o no atributos. Dado que en UML toda clase debe tener un nombre, usualmente se traslada el nombre de la asociación a la clase. Una clase asociación nunca tendrá atributos identificadores.

#### Ejemplo 4

Sea el siguiente diagrama que representa la información sobre discos y canciones. La asociación *Contiene* representa las canciones que contiene cada disco.



Si además se quiere mantener información de los cantantes, deberemos incluir una nueva clase con la información que se quiera almacenar. Por otra parte, si se quiere poder incluir la información de qué cantante canta una determinada canción en un disco, se deberá incluir una nueva asociación (*Canta*) que relacione una ocurrencia de *Contiene* (un par canción/disco) con una ocurrencia de *Cantante*. Para representar esta información, la asociación *Contiene* se debe considerar como una clase asociación, en este caso sin atributos. En la siguiente figura se muestra la modificación:



### 2.1.4 Clases débiles

Una clase sufre *restricción de dependencia de identificación* cuando no puede identificarse con sus propios atributos de manera que sus ocurrencias son distinguibles gracias a su asociación con otra/s clase/s. A este tipo de clases se les denomina *clases débiles*. Esta restricción implica siempre una restricción de existencia.

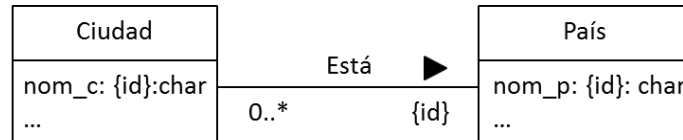


Esta restricción se representa con la etiqueta {id} en lugar de la cardinalidad (en este caso la cardinalidad es siempre 1:1), e implica que la clase débil cuenta entre sus atributos con el o los atributos identificadores de la/s clase/s de las que depende, sin que tengan que representarse explícitamente.

A continuación se muestran varios ejemplos que aclaran este concepto.

### Ejemplo 5

Sea el siguiente diagrama:

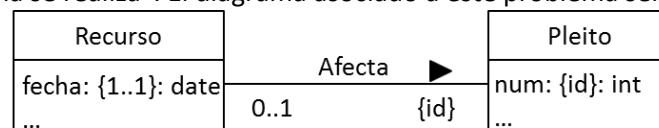


La clase *Ciudad* es débil ya que en la organización (información geográfica mundial) que se está modelando puede haber varias ciudades con el mismo nombre, aunque evidentemente siempre en distintos países; entonces ¿cómo se distingue una ciudad de otra? En primer lugar, por el país al que pertenecen y dentro de un mismo país por el atributo *nom\_c* necesario en este caso ya que en un país puede haber muchas ciudades

Un caso menos frecuente se presenta a continuación.

### Ejemplo 6

“Una organización tiene informatizados los expedientes jurídicos de todos los pleitos en los que está involucrada; de cada pleito, entre otras informaciones, se conoce el número de pleito y el resultado de la sentencia que puede ser favorable o desfavorable. En caso de ser desfavorable se puede presentar como mucho un recurso; cada recurso se identifica por el número de pleito al que atañe y entre otros atributos interesa saber en qué fecha se realiza”. El diagrama asociado a este problema sería:



Dado que un pleito se puede recurrir como mucho una vez, para distinguir un recurso de otro sólo se necesita saber qué pleito se está recurriendo no siendo necesario en este caso un atributo de la clase recurso que ayude a distinguirlos.

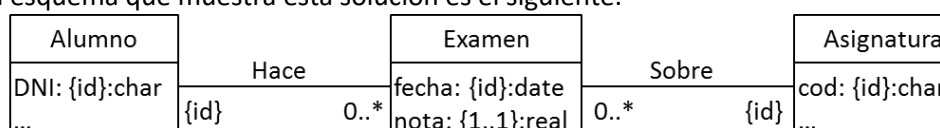
Resumiendo, las clases débiles son aquéllas para las que no existe un identificador como se ha definido antes, sino que para distinguir entre sus ocurrencias es necesario utilizar una (o más de una) de sus asociaciones siendo necesario la existencia de un atributo (o conjunto de atributos) llamado *semiidentificador* cuando la cardinalidad máxima de la otra clase sea mayor que 1.

Existe la posibilidad de que la clase se identifique gracias a varias asociaciones con otras clases; en este caso, las cardinalidades máximas de todas estas clases en esas asociaciones deben ser mayores que 1.

### Ejemplo 7

Supónganse que se quiere almacenar información sobre las notas que han obtenido los alumnos en distintas asignaturas teniendo en cuenta que un alumno se puede presentar varias veces a una asignatura, pero en fechas distintas, y que los exámenes no tienen identificador.

Una posible forma de identificar cada examen es sabiendo qué alumno lo ha hecho, de qué asignatura es y en qué fecha se realizó la prueba (para distinguir los exámenes del mismo alumno en la misma asignatura). El esquema que muestra esta solución es el siguiente.



### 2.1.5 Asociación n-arias

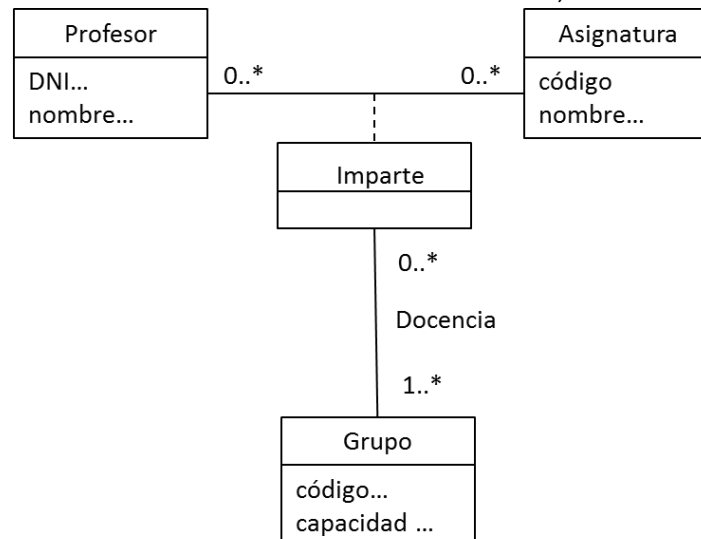
Una asociación que conecta exactamente dos clases se dice que es *binaria*; sin embargo, en el mundo real se pueden encontrar relaciones entre objetos que necesitarían ser representadas por asociaciones que conectaran más de dos clases, *asociaciones n-arias* (realmente con relaciones entre tres clases se pueden representar la mayoría de los SI). Aunque en el diagrama de clases de UML podrían representarse explícitamente este tipo de relaciones de grado mayor que dos, esta representación no se suele utilizar, utilizando para representar este tipo de relaciones o las clases asociación o las clases débiles.

#### Ejemplo 8

Sea un sistema de información sobre profesores, asignaturas y grupos de una academia que prepara a los alumnos para la obtención del título de la ESO en el que se cumplen las siguientes propiedades:

- Un profesor puede impartir cualquier asignatura, pero si imparte una asignatura lo hace en al menos un grupo.
- Una asignatura puede ser impartida por cualquier número de profesores en cualquier número de grupos.
- En un grupo se puede impartir cualquier número de asignaturas por cualquier número de profesores.

La solución usando una clase asociación se muestra a continuación;

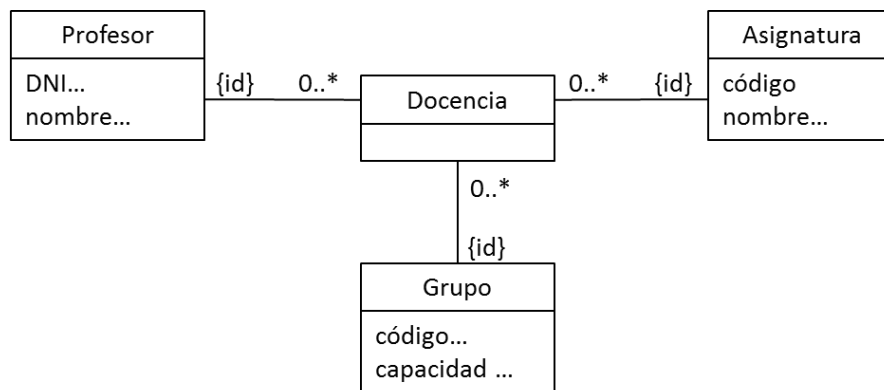


La manera de interpretar las cardinalidades es la siguiente:

- Un profesor puede impartir de 0 a N asignaturas.
- Una asignatura puede ser impartida por de 0 a N profesores.
- Cuando un profesor se relaciona con una asignatura a través de la asociación *Imparte*, se debe relacionar al menos con un grupo.
- Un grupo se relaciona entre 0 y N pares profesor asignatura (ocurrencias de la asociación *Imparte*).

Con este diseño queda representado que la relación *Docencia* es en realidad una relación ternaria ya que no se sabe qué asignatura imparte un profesor hasta que no se sabe a qué grupo o grupos está asignado.

La solución usando una clase débil se muestra en la figura siguiente:



En esta solución, la clase *Docencia* no tiene atributos propios ya que se diseña para representar la relación a tres bandas entre profesores, asignaturas y grupos.

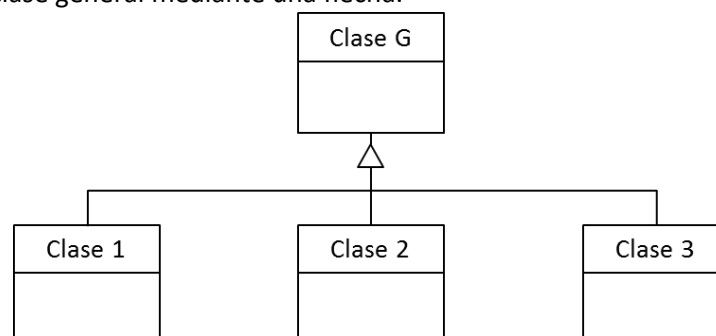
### 2.1.6 Generalización/Especialización

Cuando se detecta que entre distintas clases definidas en el esquema existe una relación de inclusión (esto es, que todas las ocurrencias de una clase son a su vez ocurrencia de otra más general), este hecho se expresa por medio de la Generalización/Especialización. Esto significa que la clase más general se especializa en una o varias clases especializadas o subclases, o dicho a la inversa, que una o varias clases se generalizan en una clase general o superclase. Este proceso se puede repetir a distintos niveles, siendo posible que una clase tenga más de una superclase, siempre que la clase más general del conjunto sea única. La clase más general será además la única que tenga identificador. Todas las subclases de una clase tienen, además de sus atributos propios, todos los atributos de sus superclases (en cualquier nivel), pero no deben representarse explícitamente en el diagrama.

Aunque en el UML se permite que una clase especializada repita un atributo de una clase superior con la idea de redefinir la característica, en el uso que se va a hacer en esta asignatura de los diagramas de clases, esto no se permite.

Una clase puede participar en distintas Generalizaciones/Especializaciones que se definen atendiendo a criterios distintos. El criterio se puede indicar al lado del arco.

La especialización se representa uniendo todas las clases especializadas según un criterio con un arco que se conecta con la clase general mediante una flecha.



La especialización de una clase en varias subclases puede ser, según el recubrimiento que hagan las clases especializadas de la clase general en:

- *Total* (o Completa): en este caso todas las ocurrencias de la clase general deben corresponderse con al menos una ocurrencia de alguna de las subclases; o
- *Parcial* (o incompleta): cuando puedan existir ocurrencias de la clase general que no se especialice en ninguna de las subclases.

Por otra parte, según la posible intersección entre las clases especializadas, puede ser:

- *Solapada*: cuando una ocurrencia de la clase general pueda corresponderse con más de una ocurrencia de las subclases; o

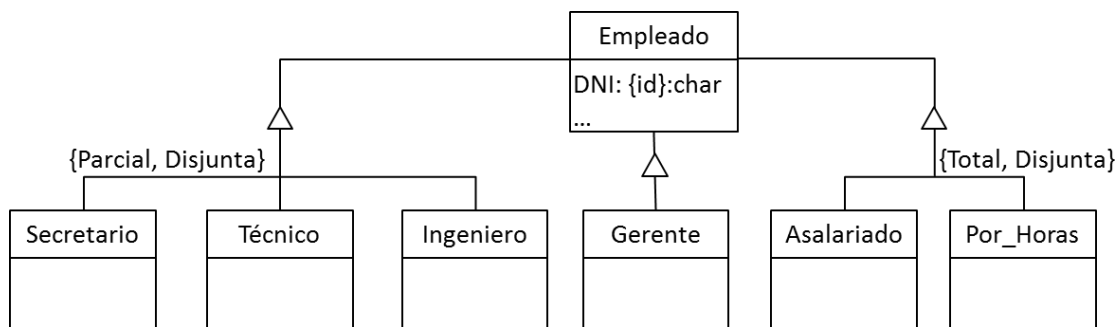
- *Disjunta*: cada ocurrencia de la clase general puede corresponderse como mucho con una ocurrencia de alguna de las subclases.

Estas propiedades se especifican entre llaves junto al arco que representa la especialización. Los valores por defecto son parcial y solapada, es decir, los menos restrictivos.

### Ejemplo 9

En el siguiente diagrama de clases se expresa que los empleados de una empresa pueden especializarse de acuerdo con tres criterios:

- Según el cuerpo al que pertenece.
- Según que el empleado sea o no gerente.
- Según el tipo de contrato que tiene.



## 2.2 Obtención del diagrama de clases

El objetivo fundamental de este apartado es enseñar cómo se puede abordar la tarea del diseño conceptual de las propiedades estáticas de un sistema de información utilizando el diagrama de clases presentado, es decir, proponer una metodología de diseño, así como destacar puntos conflictivos que pueden aparecer en el desarrollo de un sistema de información. Desafortunadamente, no es posible definir un algoritmo que permita obtener, de forma determinista, a partir de las especificaciones el mejor diagrama posible, de manera que la metodología se reduce a un conjunto de tareas a realizar y a una serie de consejos que pretenden avisar al alumno de ciertos detalles que pueden ayudarle a definir un diagrama correcto al menos sintácticamente.

Para obtener un diagrama adecuado y fiable a partir del análisis de la realidad y de los requisitos de la organización hay que realizar las siguientes actividades:

- Identificar las clases con sus atributos,
- Identificar generalizaciones/especializaciones,
- Identificar asociaciones entre clases, y
- Especificar restricciones de integridad.

Estas actividades se realizan de forma iterativa hasta conseguir definir un diagrama de clases lo más fiel posible a la realidad y en el cual el conjunto de restricciones de integridad explícitas incluidas en el anexo al diagrama sea lo más pequeño posible. A continuación, se comentan todas ellas.

### 2.2.1 Identificar las clases con sus atributos

Por cada tipo de objeto de la realidad, una vez concretada la información descriptiva que se desea almacenar, se definirá una clase en el diagrama. Una clase viene definida por un conjunto de atributos que representan la información que se desea conocer de cada tipo de objeto. Para cada atributo se debe:

- Asociar un tipo de datos, y
- Especificar la cardinalidad y las restricciones de unicidad si es que las hay.

De entre estos atributos, si es posible, se destacarán los atributos del identificador; si no existe identificador, la clase debe ser considerada débil y habrá que decidir, cuando se estudien las asociaciones,

sobre cuál o cuáles se apoya para identificarse.

No hay que pensar en que antes de avanzar en el diseño hay que definir un conjunto de clases que sea fijo, sino que éste puede cambiar a medida que se tomen ciertas decisiones de diseño. Por ejemplo, es posible que algunos atributos inicialmente considerados desaparezcan luego y se conviertan en clases.

## 2.2.2 Identificar generalizaciones/especializaciones

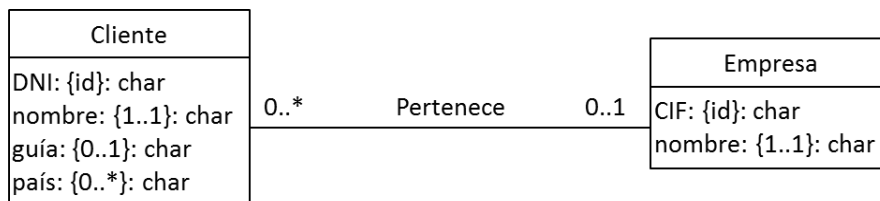
La especialización es el proceso por el que se clasifica una clase de objetos en subclases más especializadas. La generalización es el proceso inverso por el que se generalizan varias clases para obtener una abstracta de más alto nivel que incluya los objetos de todas estas clases. La especialización es un refinamiento conceptual mientras que la generalización es una síntesis conceptual.

Se pueden distinguir tres procesos mentales que pueden conducir a definir una generalización/especialización.

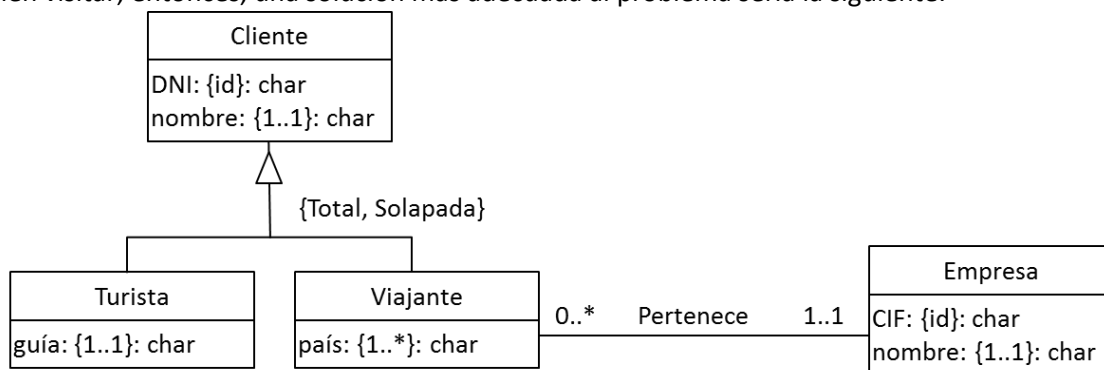
1) Estrategia descendente (especialización): en el conjunto de ocurrencias de una clase, se pueden definir subconjuntos con propiedades estáticas (atributos) o de comportamiento (asociaciones) distintas.

### Ejemplo 10

En el contexto de una agencia de viajes, sobre los clientes de la agencia, se ha diseñado el siguiente diagrama:



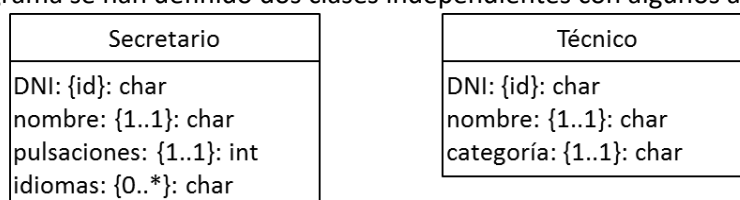
Más tarde se detecta que hay dos clases de clientes: los turistas, a los que siempre se asignará un guía; y los viajeros de negocios, que siempre pertenecen a una empresa y de los que interesa conocer los países que suelen visitar; entonces, una solución más adecuada al problema sería la siguiente:



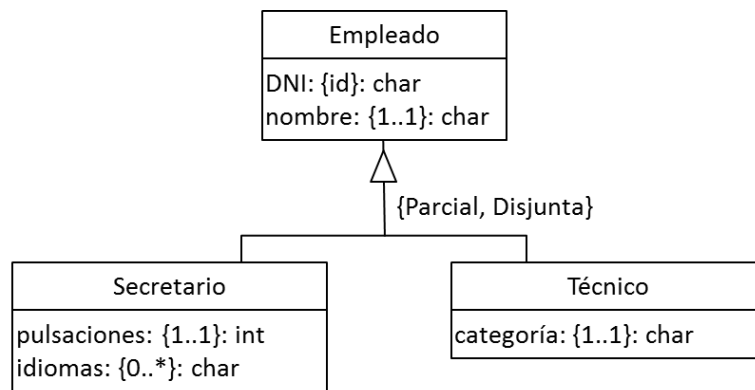
2) Estrategia ascendente (generalización): existe en el esquema un conjunto de clases con algunas propiedades similares y que en la realidad se podrían clasificar en un objeto común.

### Ejemplo 11

En el siguiente diagrama se han definido dos clases independientes con algunos atributos comunes.



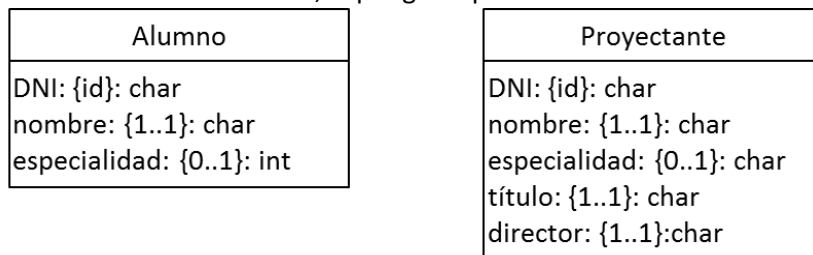
Si además se observa que ambas clases se refieren a trabajadores de la empresa que para algunos procesos conviene manipular juntos, sería más correcto considerar una clase general Empleado:



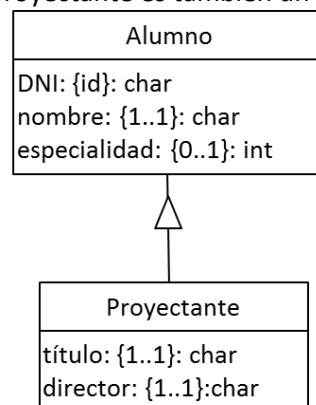
3) Jerarquía (relación *Es un*): se detecta una relación de inclusión entre clases previamente definidas.

### Ejemplo 12

En el contexto de una escuela universitaria, supóngase que se han definido dos clases:



Pero si se tiene en cuenta que todo proyectante es también un alumno, la solución más adecuada sería:

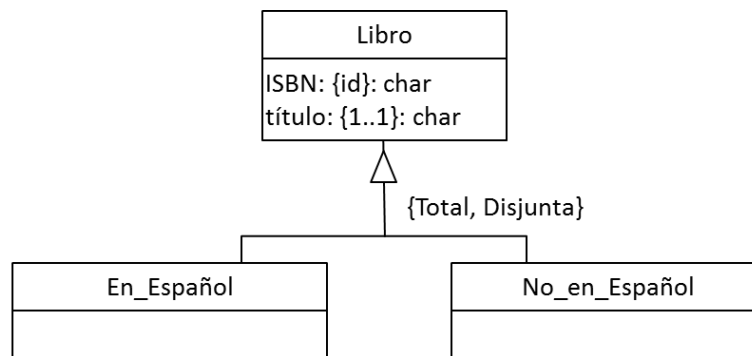


Como se puede observar en los ejemplos, en cualquiera de estos casos se ha definido una generalización/especialización de manera que los atributos identificadores y los descriptores que son comunes a todas las clases estén en la clase general, quedándose los atributos específicos y las asociaciones específicas en cada una de las clases especializadas.

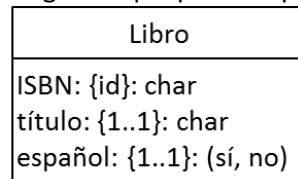
Hay que darse cuenta de que la generalización/especialización no debe definirse por los nombres que puedan tener los atributos sino cuando realmente exista entre los objetos la relación de subclase que implica este concepto. Por otra parte, una generalización/especialización en la que las clases especializadas no tienen propiedades distintivas (atributos o asociaciones) no resulta muy útil pudiéndose representar la misma información y de forma más sencilla con un atributo discriminador en la clase general (esta idea se ilustra en el ejemplo siguiente).

### Ejemplo 13

Supóngase que interesa saber si los libros de una biblioteca están escritos en español o en otros idiomas. Un diseño como el siguiente:



es demasiado complicado si la única diferencia entre los libros es si están escritos o no en español. Se puede expresar lo mismo con el siguiente diagrama que por otra parte resulta mucho más sencillo:



Se debe indicar qué tipo de generalización/especialización se está definiendo, especificando si es total o parcial y si es disjunta o solapada como se ha hecho en todos los ejemplos presentados, excepto si hay una sola subclase, en cuyo caso no hace falta, ya que siempre será parcial, y no tienen sentido las propiedades *disjunta* y *solapada*.

### 2.2.3 Identificar asociaciones entre clases

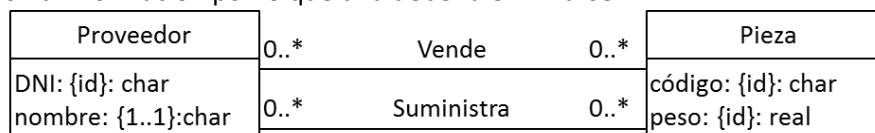
Una vez definido un conjunto inicial de clases que, como ya se ha comentado, podrá ser reconsiderado a lo largo del todo el diseño, hay que estudiar las asociaciones existentes entre ellas, ya que raramente existirán clases sin conexiones con otras. Para definir una asociación hay que especificar las clases implicadas y las cardinalidades máximas y mínimas.

Para la definición de un conjunto de asociaciones adecuado es importante tener en cuenta las siguientes indicaciones:

- 1) Ante la duda se recomienda elegir las cardinalidades menos restrictivas (0 para la mínima y \* para la máxima).
- 2) Las asociaciones redundantes deben eliminarse. Dos o más asociaciones se consideran redundantes si representan el mismo concepto; sin embargo, hay que darse cuenta de que entre las mismas clases se pueden definir más de una asociación siempre que tengan significados diferentes.

#### Ejemplo 14

En el siguiente diagrama, aunque con nombres diferentes, se han definido dos asociaciones que representan la misma información por lo que una debería eliminarse:



Sin embargo, pueden existir dos asociaciones definidas sobre las mismas clases pero con significados completamente distintos:

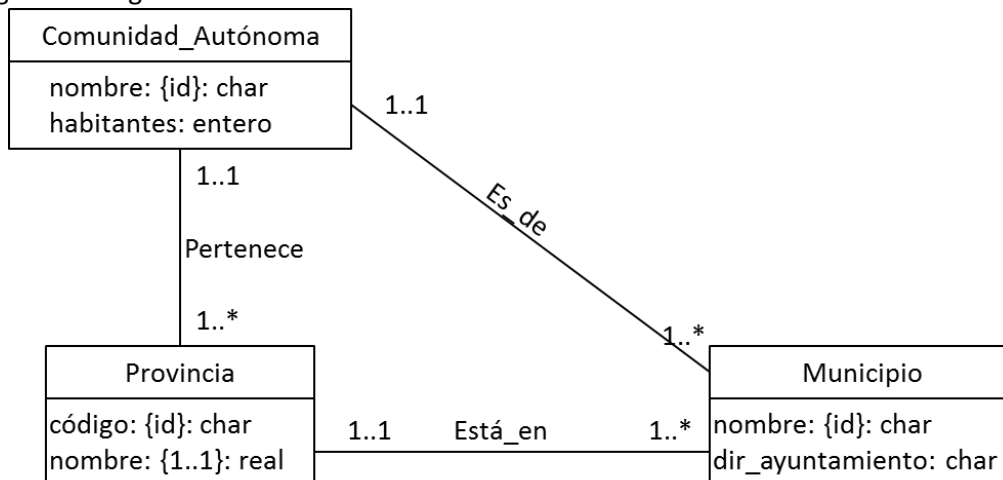


- 3) Eliminar la redundancia que se deriva de dependencias transitivas, una asociación será redundante por dependencias transitivas cuando pueda derivarse a través de otras asociaciones (dos o más). En caso de

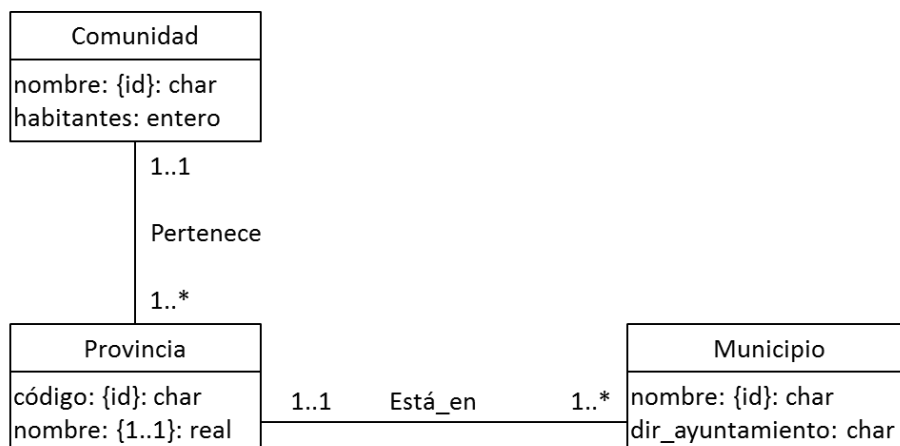
desea mantener la redundancia, deberán especificarse propiedades que eviten la inconsistencia de la información.

### Ejemplo 15

En el siguiente diagrama se han definido tres asociaciones:

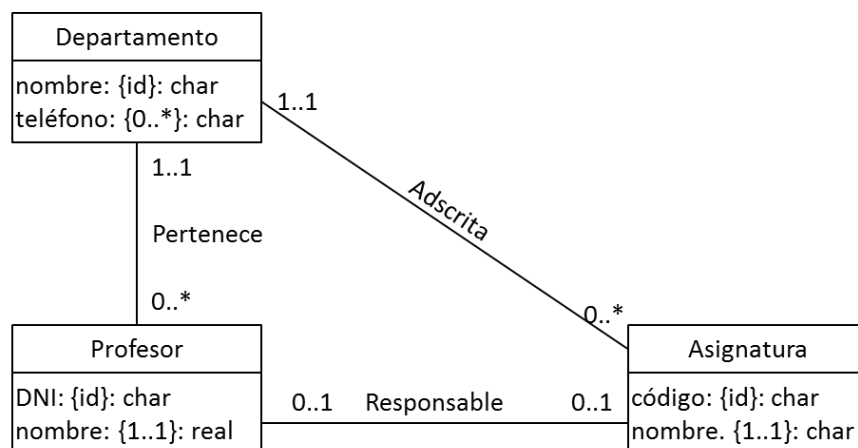


La asociación *Es\_de* es redundante ya que sus ocurrencias se pueden derivar a partir de *Pertenece* y *Está\_en* (un municipio es de la comunidad autónoma a la que pertenece su provincia); por ello se debería eliminar.



Sin embargo, no siempre es posible eliminar la redundancia como se muestra en el siguiente ejemplo.

### Ejemplo 16



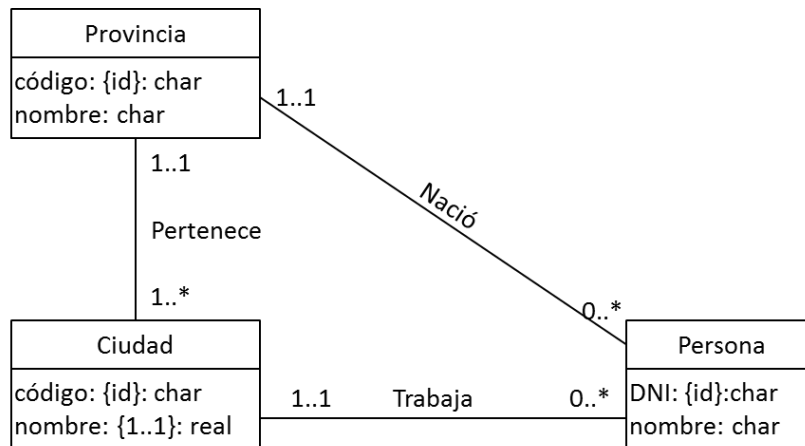
Si en este sistema de información se debe cumplir que los profesores sólo son responsables de asignaturas de su departamento, entonces el departamento al que pertenece un profesor puede derivarse a través del departamento al que está adscrita la asignatura de la que es responsable, pero como puede



darse el caso de que un profesor no sea responsable de ninguna asignatura no se puede eliminar. La misma reflexión puede hacerse respecto a la asociación *Adscrita*. Así pues, dado que pese a existir cierta redundancia no es posible eliminar ninguna asociación sin perder por ello información, este diagrama necesita una restricción de integridad que controle la restricción: “los profesores sólo pueden ser responsables de asignaturas de su departamento”.

Sin embargo, no hay que pensar que siempre que hay un ciclo entre clases existe una dependencia transitiva, como se ilustra en siguiente ejemplo.

### Ejemplo 17

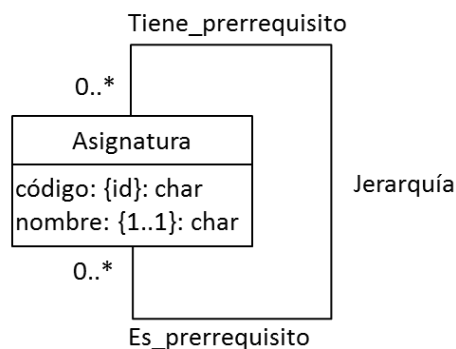


En este caso, pese a existir un ciclo de las mismas características que en el Ejemplo 15 no existe redundancia ya que una persona no tiene por qué haber nacido en la misma provincia en la que está la ciudad en la que trabaja.

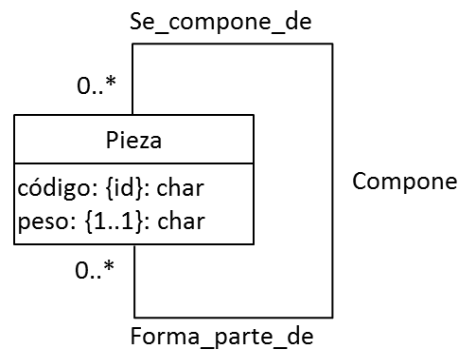
4) Especificar el rol que cada clase juega en una asociación cuando alguna clase participa más de una vez en ella. El caso más sencillo se presenta en las asociaciones en las que las dos clases relacionadas son la misma. En el siguiente ejemplo se muestran algunas asociaciones de este tipo especificando el rol o papel.

### Ejemplo 18

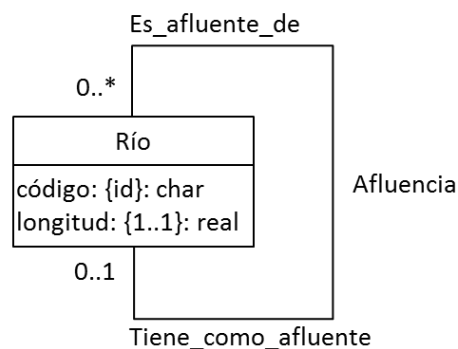
1) Relación de prerequisites en el conjunto de las asignaturas de una carrera: “una asignatura puede ser prerequisite de muchas asignaturas y tener también a muchas asignaturas como prerequisite”.



2) Relación de composición entre piezas: “una pieza se compone de muchas piezas y a su vez puede formar parte de muchas piezas”.



3) Relación entre los ríos por el hecho de que unos son afluentes de otros: “un río puede ser afluente de otro, pero a su vez muchos ríos pueden afluir a él”



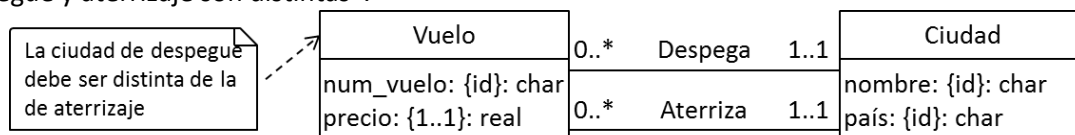
En general, hay que ser cuidadosos con las asociaciones reflexivas ya que normalmente exigen que se especifiquen ciertas propiedades que no quedan contempladas en la definición de la asociación. Por ejemplo, las tres asociaciones representadas en el ejemplo anterior son antisimétricas (una asignatura no puede ser prerrequisito de un prerrequisito suyo) y antirreflexivas (una asignatura no puede ser prerrequisito de sí misma), propiedades que no se expresan en el diagrama.

## 2.2.4 Especificar restricciones de integridad

Para terminar con el diseño, todas aquellas propiedades de la realidad que no hayan quedado expresadas en el diagrama de clases deben incluirse. Para ello puede hacerse uso de los *elementos de anotación* de los diagramas de UML. Los elementos de anotación son comentarios que se pueden aplicar para describir, clarificar y hacer observaciones sobre cualquier elemento del modelo. Gráficamente se representa con un rectángulo con una esquina doblada en el que se incluye un texto. El rectángulo se conecta con una flecha discontinua apuntando hacia el objeto al que se refiere.

### Ejemplo 19

Para el diagrama del ejemplo 14 se podría incluir la siguiente restricción: “Para todo vuelo las ciudades de despegue y aterrizaje son distintas”:



## 2.2.5 Otras consideraciones

Para finalizar, en un diagrama de clases correcto hay que tener en cuenta las siguientes reglas:

- No puede haber dos nombres iguales entre las clases, las asociaciones ni los roles.
- Una clase siempre tiene identificador excepto cuando la clase está en uno de los tres casos siguientes:
  - Es débil, o

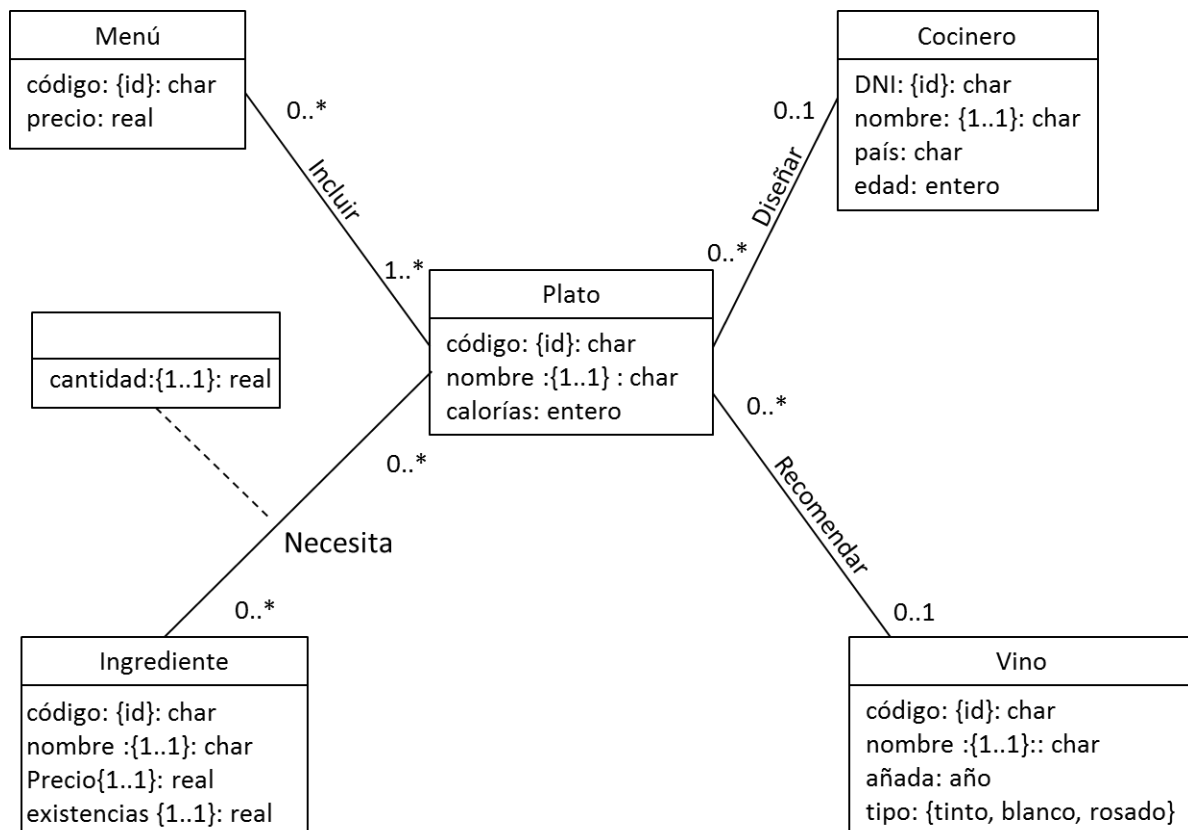
- Es una especialización, o
- Es una clase asociación.
- Las asociaciones nunca tienen identificador.
- Los nombres de atributos en una clase no se repiten.
- Las clases especializadas no pueden tener un atributo con el mismo nombre que algún atributo de sus clases generales.
- Las clases especializadas nunca tienen identificador ni son débiles ya que heredan la identificación de su clase general.
- En el diagrama de clases no existen las claves ajenas (concepto propio del modelo relacional) de forma que nunca se debe incluir un atributo en una clase con la intención de que represente una asociación con otra clase.

### 3 EJERCICIOS RESUELTOS

#### 3.1 Restaurante

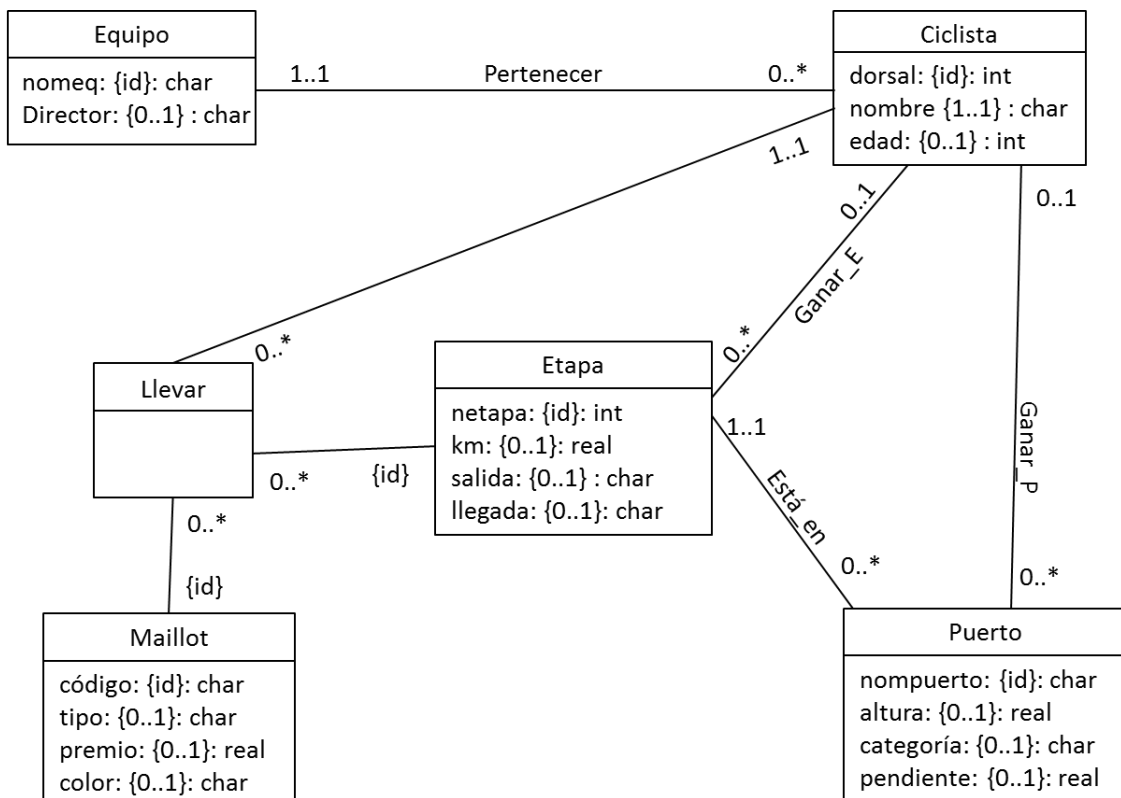
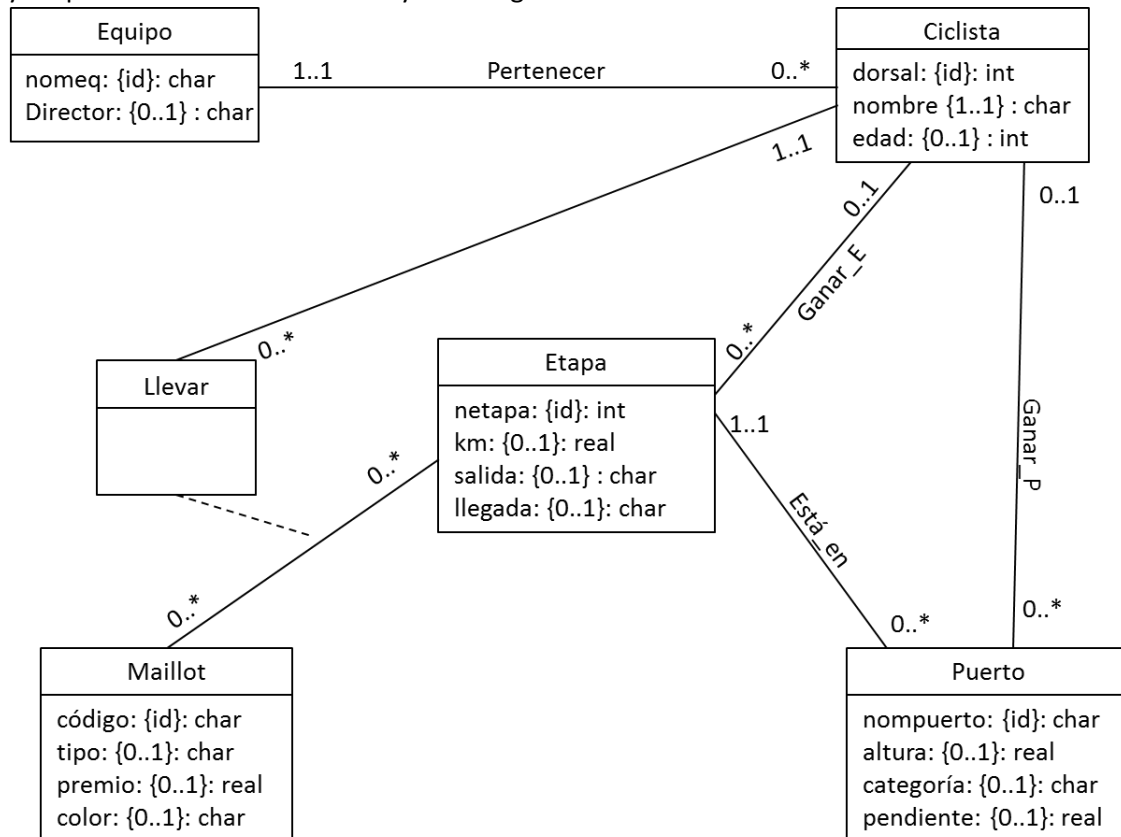
“Un restaurante desea que sus menús puedan ser consultados desde terminales y para ello ha decidido diseñar una base de datos relacional. En el restaurante se ofertan varios menús de los que se quiere saber el código, el precio y los platos que lo componen (al menos uno); cada plato puede haber sido diseñado como mucho por un cocinero del restaurante (que tiene DNI y nombre, también puede tener país de origen y edad como datos de interés); de los platos se debe saber el código asignado al plato, el nombre y quizás las calorías aproximadas; también puede interesar almacenar para algunos platos los ingredientes necesarios (indicando la cantidad) y el vino que se recomienda para su completo disfrute. De cada vino disponible en el restaurante se quiere conocer su código (interno al restaurante) y el nombre, también se desea conocer el tipo, y la añada. Por último y para poder controlar la despensa, de cada ingrediente se debe almacenar un código, el nombre, el precio en mercado y las existencias que quedan

El diagrama de clases que representa este sistema de información es el siguiente:



### 3.2 Ciclismo

Los siguientes diagramas de clases representan el sistema de información de la base de datos de ciclismo, utilizada en práctica. En el primero se plantea la representación de la relación entre maillot, ciclista y etapa con una clase asociación y en el segundo con una clase débil.



## 4 EJERCICIOS PARA RESOLVER

A continuación, se describen varios sistemas de información para los que hay que realizar el diseño conceptual utilizando el diagrama de clases de UML.

### 4.1 Una compañía de seguros

Una compañía de seguros desea diseñar un sistema de información para gestionar las peritaciones de los vehículos accidentados a su cargo. Cada peritación se identifica con un número de referencia y se debe conocer necesariamente la fecha de realización, el perito asignado (código y nombre), los datos del taller donde se ha realizado (nombre del taller y domicilio) y del vehículo peritado. Cada vehículo, se codifica con un identificador secuencial. Para los vehículos matriculados además se almacena la matrícula, para los ciclomotores el número de placa municipal y para cualquier otro (por ejemplo, bicicletas) un código interno (las matrículas, números de placa, y códigos internos en conjunto deben ser únicos). Además, interesa saber la marca, modelo, propietario y si está asegurado, el número de póliza, la compañía, el tipo y la fecha de caducidad (toda póliza siempre va ligada a un vehículo). Sólo se tiene información de los vehículos sobre los que se hacen peritaciones.

Se dispone de un catálogo de las diferentes partes de los vehículos codificadas con su correspondiente descripción (por ejemplo, XX1 Aleta delantera, XX2 Puerta Derecha, XX3 Faro Derecho, etc.). El resultado de la peritación consiste en una estimación de las partes del vehículo afectadas. Para cada parte afectada, se emite un diagnóstico en el que se indica si se ha de reparar o sustituir, así como el tiempo de mano de obra estimado. Además, se detallan los materiales a utilizar en la reparación de cada parte afectada. De estos materiales también se dispone de un catálogo con el código, descripción y precio.

### 4.2 Una galería de pintura

Una galería de pintura desea diseñar una base de datos para almacenar la información sobre las exposiciones que organiza y los cuadros que vende. La información más relevante que desea almacenar es la que se describe a continuación.

Con la galería trabajan distintos artistas de los que se quiere saber: DNI, nombre, dirección y teléfonos. Estos artistas proporcionan pinturas a la galería para que ésta organice exposiciones con el propósito de venderlas. Cada pintura se identifica por un código, tiene un título y un precio fijado por el artista. Las exposiciones se identifican por un código y tienen una fecha de inicio y otra de finalización. También se quiere tener información de clientes, a los que se avisará de las distintas exposiciones para que acudan a verlas y adquieran, si así lo desean, alguna obra; de estos clientes se desea conocer el DNI, el nombre, la dirección y los teléfonos. Los artistas también pueden ser clientes, por lo que también se les avisa de las exposiciones.

Una pintura puede exhibirse en varias exposiciones hasta que se consiga vender a algún cliente; se desea mantener la información de qué pinturas se han incluido en cada exposición; cuando se venda un cuadro, hay que almacenar la fecha de la venta, que no tiene por qué coincidir con una exposición, y el precio final que se ha pagado.

### 4.3 Biblioteca

Se desea diseñar una base de datos para la gestión de una pequeña biblioteca de un departamento. En el análisis del sistema se han identificado los requerimientos que van a hacerse con más frecuencia; éstos son:

- Consultar los datos de un libro: código del libro, título, autor (o autores), temática y en caso de estar prestado, el socio que lo tiene actualmente en préstamo.
- Consultar la información sobre un socio: código del socio, nombre, dirección, teléfono y libros que actualmente tiene en préstamo, así como la fecha de préstamo.

- Dar de alta, dar de baja y modificar los datos de un socio.
- Gestionar los préstamos: prestar un libro a un socio y registrar la devolución de un libro.

Algunas restricciones de integridad que se han detectado son:

- El código del libro identifica unívocamente al libro.
- El código del socio identifica unívocamente al socio.
- Los temas utilizados para clasificar un libro son: física, electricidad, mecánica y óptica.
- El número total de libros que tiene prestados un socio es un *dato derivado* que será mantenido automáticamente por el sistema.

#### 4.4 La revista “La Periódica”

La redacción de LA PERIÓDICA, revista quincenal de información económica y política, necesita un sistema de base de datos, para almacenar la información de los números que publica, los artículos que en ellos aparecen, y otros aspectos relacionados, que se explican a continuación.

Como hemos dicho, se publica un número de la revista cada quince días. Cada número tiene, como atributos, un identificador, una fecha de publicación y una tirada (expresada en cantidad de ejemplares). Un número contiene diversas secciones (algunas de ellas pueden ser: economía, actualidad, política nacional, política internacional, inversiones, ...) donde aparecen los artículos publicados, las cuales se identifican por un código de sección y tienen una descripción.

Cada artículo se identifica por un código, y además tiene un título, un resumen y el texto a publicar. Cuando se da de alta un nuevo artículo, se debe conocer quién es el autor (su código, su nombre y su teléfono de contacto), y se debe decidir en qué sección va a aparecer. En un momento posterior, se elige el número de la revista donde se publicará. Un artículo sólo aparecerá en un número.

Los artículos son de dos tipos, según el autor sea un miembro del equipo de redacción de la revista, o bien un colaborador externo. Las primeras se llaman crónicas, y tienen un atributo que refleja la urgencia de publicación. Las segundas, o colaboraciones, tienen una valoración y una justificación de esta valoración.

Las colaboraciones siempre llegan por correo electrónico, por lo que se desea tener un registro de los mensajes recibidos, donde cada entrada tendrá un número de orden, la fecha de entrada, el remitente y el asunto del mensaje. Con cada mensaje, llega un único documento de texto, que puede ser de dos tipos. Puede ser el documento original, con el texto para componer una nueva colaboración, o bien será un documento complementario, para realizar alguna modificación en el texto de una colaboración que ya existe. En este último caso, se sabrá el tipo de modificación requerida (que puede ser: añadir el texto del documento complementario al texto de la colaboración, sustituir parte del texto de la colaboración por otro texto, eliminar la parte del texto de la colaboración que aparece en el documento complementario). De una colaboración, habrá un y sólo un documento original, y se podrán recibir varios documentos complementarios.

#### 4.5 Una compañía de alquiler de vehículos (Prácticas)

Una compañía de alquiler de vehículos desea informatizar sus actividades principales.

De los vehículos que se alquilan se conoce su matrícula, nº de bastidor, marca, modelo, fecha de fabricación, kilometraje y nº de ocupantes. Existen cuatro categorías de vehículos en la compañía: turismo, lujo, todo terreno y furgoneta; dependiendo de la categoría del vehículo se aplican diferentes tarifas en el precio del contrato de alquiler.

Los clientes que alquilan los vehículos pueden ser personas físicas, o bien entidades de algún tipo (empresas, organismos públicos, etc.). Para cualquiera de ellos se desea conocer su NIF, dirección y teléfono. En caso de tratarse de una persona se desea conocer su nombre (nombre propio y apellidos), edad y años que posee el carné de conducir; si se trata de una entidad se desea conocer el nombre de la misma y el nombre de la persona de contacto.

Se dispone también de un conjunto de chóferes para los alquileres con conductor. De cada chófer se conoce su D.N.I., nombre, dirección, teléfono, fecha de alta, nº de la seguridad social y salario. Cuando se realiza un contrato de alquiler se especificará si incluye o no chófer y, en caso afirmativo, qué chófer es el asignado. Claramente un chófer sólo puede estar asignado en un cierto instante a un único contrato de alquiler que no haya finalizado.

El contrato de alquiler permite saber qué persona o entidad ha alquilado un cierto vehículo. De cada contrato se conoce su nº de contrato, fecha de inicio (que es obligatoria) y fecha de devolución (teórica y real). El contrato puede incluir chófer y debe especificar si se pagará por kilómetros, días o semanas. La forma de calcular el importe del contrato es la siguiente: según sea el contrato por km., días o semanas se aplica la tarifa del tipo de vehículo alquilado y si se incluyó chófer se incrementa con la parte correspondiente al mismo. Esto queda resumido a continuación:

- $\text{nº kilómetros} * \text{tarifa\_kilometros\_categoría\_vehículo}$
- $\text{días} * \text{tarifa\_días\_categoría\_vehículo}$
- $\text{semanas} * \text{tarifa\_semanas\_categoría\_vehículo}$
- $\text{días} * 10.000 \text{ ptas. (si incluye chófer)}$

Además, por cada día de retraso en la devolución del vehículo se paga una multa de 10.000 ptas.

Las facturas incluyen uno o más contratos correspondientes a una misma persona o entidad. Las facturas tienen un número y una fecha e incluyen el conjunto de contratos de alquiler y el importe correspondiente de cada uno de ellos. Los requerimientos de consulta y actualización más frecuentes son los siguientes:

- Listar los contratos ya finalizados indicando vehículo alquilado y datos del cliente
- Realizar la factura
- Listar los vehículos disponibles en un momento dado
- Listar las tarifas para cada tipo de vehículo

## 4.6 Un supermercado (Prácticas)

Una gran tienda de comestibles ha decidido, para ser competitiva, repartir a domicilio la compra de sus clientes. Para ello ha habilitado un almacén junto a la tienda donde se depositarán las compras hasta que un mozo (o moza) las recoja para repartirlas. En este almacén se han instalado varias cámaras de frío que se han codificado de forma única y de las que interesa saber la fecha de revisión así como el nombre y teléfono del técnico encargado de su mantenimiento. En el almacén también se han instalado unas estanterías numeradas de las que interesa saber a qué altura del suelo están.

Cuando un cliente finaliza su compra debe proporcionar al cajero que la atiende sus datos (D.N.I., nombre, teléfono y dirección) si éstos no están ya almacenados. El cajero se encarga de distribuir la compra en bolsas y debe introducir en el sistema la siguiente información: el número del tique (que identifica una compra), a fecha y, para cada bolsa que forme parte de la compra, el número secuencial asignado a la bolsa (único dentro del número de tique) y el tipo de bolsa que es (de frío o no).

Estas bolsas se trasladan al almacén y se depositan en las cámaras y en las estanterías según sean de frío o no. El encargado del almacén es el que introduce en el sistema dónde está situada cada bolsa. Los mozos de reparto pasan por el almacén y recogen la nota de una compra en la que se indica la situación de todas las bolsas de dicha compra. En esta nota también se indica quién ha sido el cajero encargado de esa compra para que se le pueda consultar si surgen problemas.

Es fundamental saber la hora a la que se ha entregado cada compra, así como el mozo que la ha realizado. Esta información que la introducirá el mozo cuando vuelva, se utilizará para repartir beneficios al final de cada mes.

De cada mozo se debe almacenar el número de trabajador, el nombre, la edad y el número del teléfono móvil. De los cajeros interesa conocer el número de trabajador, el nombre y la categoría.

## 4.7 El club de vela de la UPV

Se desea diseñar una base de datos para la gestión de la bolsa de Tripulantes del club de vela de la U.P.V. La bolsa de tripulantes permite poner en contacto a los miembros de la U.P.V. (personal, alumnos y antiguos alumnos) que quieren participar en regatas, con los armadores de barcos que necesitan de tripulantes.

Para cada barco hay que almacenar sus características típicas, como el modelo, el nombre del barco que sirve para identificarlo, eslora, manga, club náutico al que pertenece y el armador. De los armadores, o propietarios de los barcos es necesario conocer su nombre completo, D.N.I., su dirección postal para enviarles circulares y uno o varios teléfonos de contacto (uno como mínimo). Por otra parte, el club de vela dispone de información de todas las regatas a celebrar, en la cual figura el nombre de la regata, el club náutico que lo organiza y la categoría de ésta. Normalmente, las regatas se repiten año tras año, convirtiéndose en una tradición. Para cada una de estas regatas en cada año se desea saber la fecha de inicio y de fin, y el número de plazas que se ofrecen en cada barco para los tripulantes de la bolsa de la U.P.V. Esta información es transmitida a los miembros de la bolsa y de esta manera se asignan tripulantes a cada uno de los barcos.

Para realizar la memoria final del club de vela será necesario mantener la información referente a cuáles son los miembros de la bolsa que han participado en cada una de estas regatas en un año concreto y en qué barcos lo han hecho. De los participantes en la bolsa es necesario conocer el número de su tarjeta federativa, su nombre y dirección, teléfono de contacto, si son personal o alumnos, y en este caso a qué escuela pertenecen, también es interesante conocer si poseen alguna titulación náutica. Para facilitar el desplazamiento a cada una de la regatas se mantendrá información de cada uno de los clubes náuticos que organizan las regatas, de esta manera se tiene su nombre, las siglas que lo identifican, dirección, teléfono, y una breve descripción.

Los requerimientos del sistema son pues:

- Listar los armadores que tienen sus barcos en la bolsa para enviarles correo.
- Listar los miembros de la bolsa para enviarles correo.
- Los barcos que necesitan tripulación en una prueba determinada.
- Cuál ha sido la participación, qué tripulantes han ido a cada regata y en qué barco cada año.
- Barcos que han ofertado plazas en un año concreto de una regata.

## 4.8 Una escuela de enseñanza obligatoria

La organización es una escuela de educación primaria y secundaria.

Dentro de cada curso, que se identifica por el número y el nivel, los grupos se codifican con una letra (A, B, C,...).

De cada grupo se conoce siempre el aula que tiene asignada, los diferentes profesores que imparten docencia en ese grupo (como mínimo uno) y, a medida que se van matriculando, los alumnos que están asignados al mismo (cada alumno está asignado a un grupo y sólo a uno). Además, cada grupo tiene necesariamente un profesor tutor que será uno de los que impartan docencia en él.

De los profesores se conoce su D.N.I. (es el identificador) y el nombre, y, de algunos, la especialidad.

De los alumnos se conocen todos los datos siguientes: nombre, número de matrícula, así como el D.N.I., nombre, dirección y teléfono de algún responsable de éste (el padre, la madre o un tutor, indistintamente). Algunos de los alumnos se quedan a comer en el colegio o van a casa en autobús escolar. De los que comen en el colegio se guarda el turno de comedor y si tienen alguna restricción alimenticia; de los que van en autobús escolar se guarda el número de autobús en el que viajan y la parada.

También se desea conocer las actividades extraescolares a las que están apuntados los niños; lógicamente un alumno puede estar apuntado en más de una actividad. Las actividades están codificadas y de ellas se guarda una descripción y los días de la semana que se llevan a cabo.



Cada curso tiene un representante de los padres, que será uno de los padres o tutor de los niños de ese curso. Cada padre sólo puede ser representante de un curso.

La escuela también dispone de material de apoyo a la docencia (material deportivo, instrumentos musicales, material para manualidades, etc.); este material está codificado y para cada artículo se guarda una descripción y el número de unidades disponibles. Al comienzo de curso, a cada profesor se le asigna el material de apoyo que necesite para cada uno de los grupos en los que imparte docencia, guardando información sobre el número de unidades de cada material que se le asignan por grupo.

## 4.9 Parque-Acuario

Se necesita diseñar una base de datos para mantener el sistema de información de un gran parque-acuario. En él existen diversos espacios, que tienen un código, un nombre único, una situación en el parque y la superficie. De los espacios, unos son peceras para tener a los animales, y otros tienen diferentes funciones (recepción, restaurante, mantenimiento, etc.) que se deben conocer para cada uno.

La información sobre los animales que ocupan el acuario se hace por especies. De cada una se conoce necesariamente, además del código que la identifica, el nombre científico (no se repite), el lugar de origen, la clase animal (mamífero, ave, pez, ...) a la que pertenece. También se guarda el número de ejemplares que hay, y en qué pecera del parque está alojada cada especie (todos los ejemplares de una especie están en la misma).

De las especies consideradas 'grandes', hay una cantidad reducida de ejemplares, y se mantiene información individualizada de cada uno. Se identifican dentro de su especie por un número, pueden tener un nombre (no repetido) y se debe saber si se ha traído o ha nacido allí, fecha de incorporación y el peso.

De las personas que trabajan en el parque se guarda el nº de empleado, que será el identificador y, necesariamente, el nombre, NIF (que es único) y sueldo. El personal está organizado según una estructura jerárquica, de manera que cada uno de los empleados es subordinado de otro empleado (su jefe) y sólo de uno. Se debe guardar la información sobre esta jerarquía, junto con la fecha en la que se asignó el jefe a cada uno. Lógicamente, una persona no puede ser subordinada y jefe de la misma persona (excepto uno, el jefe superior, que es jefe de sí mismo). Entre el personal, de los que tienen como cargo 'cuidador' se debe guardar la información sobre qué especie o especies cuida cada uno.

Otra información que se debe almacenar es la lista de precios de visita al parque. Esta lista incluirá el precio según el tipo (de adulto o de niño) y el día (festivo o laborable).

## 4.10 Una empresa de construcción

Una gran empresa de construcción se dedica a la reforma de inmuebles. Para llevar la gestión de las reformas se quiere diseñar una base de datos. De los empleados de la empresa se guarda el número de empleado, que es único, el número de seguridad social, que también es único, el nombre, la dirección, el teléfono y su función en la misma (arquitecto, aparejador u obrero).

Cuando un cliente solicita una reforma, se guardan todos sus datos, DNI, que lo identifica, nombre, dirección completa y teléfono de contacto y se registran los datos de la reforma: un código único, las especificaciones de la misma y la dirección completa del inmueble a reformar. A la reforma se asigna un arquitecto, que será el encargado de realizar el proyecto de reforma. De los arquitectos se conoce también el número de colegiado, que es único, y su especialidad.

Cuando el arquitecto realiza un proyecto de reforma, se anota que la reforma ya tiene proyecto, se registra una copia del proyecto y se asigna una fecha prevista de inicialización de la reforma.

Cuando se inicia una reforma, se registra como iniciada, se anota la fecha de inicio y se asigna un aparejador que será el encargado de la supervisión de la misma y del que se guarda, además, su número de móvil. Para las reformas iniciadas, todos los días se genera un parte de obra, en el que se detallan los obreros que trabajarán ese día, y para cada uno de ellos, el trabajo que tiene asignado. Los trabajos a realizar por los obreros están catalogados y tienen un código único y una descripción. De cada obrero, se

guarda, además, el conjunto de trabajos para los que está cualificado. Evidentemente, a un obrero sólo se le podrán asignar trabajos para los que está cualificado. Por otra parte, el parte de obra diario también incluye una relación de los materiales que se emplearán ese día, así como la cantidad necesaria de cada uno de ellos. Existe un registro de los materiales, con un código único, una descripción y sus existencias.

Al final del día, cada obrero puede comunicar al aparejador las incidencias ocurridas en el trabajo que tenía asignado. Esta información se registra en el parte de obra diario.

Una vez concluida la reforma, se registra la fecha de finalización de la reforma

#### 4.11 Una empresa de desguace

Una empresa de desguace y reparación de vehículos viejos desea informatizar sus oficinas. La descripción de su sistema de información es la siguiente:

Los vehículos con los que se trabaja en la empresa llegan, la mayoría, en alguna grúa de su propiedad. Posteriormente se decide si se desguazan o se reparan. A los vehículos que se destinan a reparación se les asigna un mecánico, que será el que la realizará.

Cada vehículo se identifica por su matrícula y tiene como atributos necesarios el año de fabricación, el modelo, el precio que se ha pagado por él y la fecha de entrada en la empresa. Si un vehículo entra varias veces en la empresa (por ejemplo, primero se arregla, se vende, y más tarde vuelve a entrar para desguace), sólo se mantiene la información de la última estancia.

En la empresa trabajan, entre otros, empleados dedicados a la reparación de vehículos (los mecánicos) y empleados dedicados a la conducción de grúas (los conductores). De todos los empleados es necesario saber el D.N.I., el número de la seguridad social que no se repite, el nombre, la dirección, el teléfono y el sueldo. Cada conductor cobra un extra según cuántos vehículos ha traído, por lo que es necesario conocer esa información. Es importante también saber en qué grúa lo ha hecho. De cada grúa, además de un identificador, se quiere saber el año en que se adquirió, el último mecánico de la empresa que la revisó y en qué fecha lo hizo. Toda grúa de la empresa se revisa cuando se adquiere, y en ocasiones posteriores.

De cada vehículo desguazado se quiere saber la fecha en que se ha acabado el desguace y los quilos de metal reciclable que se han sacado. Además, del vehículo también se pueden obtener piezas que se van a reutilizar, que se numeran secuencialmente (empezando desde el número uno para cada vehículo). De cada pieza obtenida se especificará una breve descripción. Estas piezas se pondrán a la venta, por lo que es necesario que tengan un precio.

De cada vehículo reparado se quiere saber un precio de venta estimativo; también se guarda información de la fecha en que se terminó de reparar, y en qué plaza está mientras no se vende. Las plazas de aparcamiento de coches en venta se identifican por un número.

#### 4.12 Control de llamadas (Prácticas)

Una gran empresa ha decidido desarrollar un sistema informático que le permita controlar el gasto de las llamadas telefónicas que realiza. La empresa tiene contratos con diferentes operadoras de telefonía. Cada una de ellas sólo ofrece un tipo de servicio: llamadas a teléfonos *fijos*, llamadas a *móviles* y servicios por *cable*. De cada una de ellas se conoce el N.I.F. (que las identifica), el nombre comercial, que es único, y la dirección completa. Para las empresas que ofrecen llamadas a teléfonos fijos se conoce el prefijo a marcar y un conjunto de teléfonos de llamada gratuita. Para las de servicio por cable el precio de la tarifa plana para Internet. Por último, para las de llamadas a móviles se almacena una tabla que indica el descuento que ofrece en sus llamadas dependiendo del día de la semana.

De cada llamada que se realiza se debe almacenar necesariamente la fecha, teléfono origen, teléfono destino, duración, operadora a través de la cual se ha realizado, tipo de llamada y franja horaria de la misma. Todas las llamadas se codifican con un número que es único, pero sólo dentro de cada operadora. Los diferentes tipos de llamada también están codificados, tienen un nombre (interna, local, provincial, internacional, ...) y una descripción. Las diferentes franjas horarias (por ejemplo, de 18h a 8h, de 8h a 14h,

...) son comunes a todas las operadoras, y también están almacenadas, conociéndose su código y descripción.

Para todas las operadoras se dispone de unas tarifas que indican el precio por minuto de diferentes tipos de llamadas en diferentes franjas horarias. Esta información también se desea almacenar en el sistema.

Con el fin de estudiar la posibilidad de sustituir las llamadas internas por el correo electrónico, de éstas se almacenará también los D.N.I. y nombres del empleado que la ha realizado y del que la ha recibido.

#### 4.13 La compañía “SoftWWW” (Prácticas)

La compañía SoftWWW se dedica a la realización de proyectos de informatización para grandes empresas. Cuando una empresa solicita sus servicios, se formaliza un contrato en el que se registra el código, la fecha de contrato, el N.I.F. y domicilio social de la empresa además de un informe con los requerimientos del proyecto a realizar. Posteriormente, un equipo de expertos de la compañía se reúne para elaborar la planificación del proyecto a partir de los datos del contrato. Cada proyecto se codifica, se establece una fecha de inicio y de finalización, se asigna un analista responsable y se establecen un conjunto de tareas a desarrollar. De cada tarea, que se identifica con un número secuencial dentro de cada proyecto, se almacena su descripción, el tiempo estimado de realización y se asigna una serie de programadores encargados de llevarla a cabo. En algunos casos, ciertas tareas requieren la utilización de alguno de los ordenadores especiales del gabinete que tienen instalado un software específico. En estos casos, se reserva por un periodo de tiempo un ordenador para que lo pueda utilizar un programador asignado a una tarea que requiera esos recursos. Los ordenadores se hallan convenientemente codificados y tienen un nombre y una ubicación. De los empleados se dispone de información de sus datos personales: D.N.I., nombre, domicilio, teléfono y puesto de trabajo que ocupan en la compañía (analista, programador, administrativo, etc.).

#### 4.14 Una compañía de líneas aéreas

Una pequeña compañía de líneas aéreas desea establecer una base de datos respecto a lo que se conoce como “vuelos genéricos”, que son vuelos diarios entre dos ciudades siempre a la misma hora y con un código determinado. Puede haber más de un vuelo genérico entre las mismas ciudades siempre que sean a diferentes horas.

En un día concreto, para cada vuelo, se desea conocer el avión asignado, el aeropuerto origen y destino, y la capacidad del vuelo. Además, se debe guardar la información necesaria sobre la reserva de plazas: lista de pasajeros (con su nombre es suficiente), número de plazas libres y tripulación asignada. En cuanto a este último punto, hay que tener en cuenta que siempre habrá como mínimo un comandante y un auxiliar de vuelo. Del personal que compone la tripulación de los vuelos, además de los datos usuales (nº empleado, D.N.I., nombre, categoría, etc.) es fundamental conocer un número de teléfono donde se pueda contactar con ellos para suplir bajas inesperadas.

De la flota de aviones de la compañía interesa conocer para cada uno: el año de fabricación y la matrícula y el número de asientos. También existe información sobre las ciudades en las que hay aeropuertos, que se desea almacenar, como es el nombre, el país al que pertenece, la temperatura media para cada estación, el número aproximado de habitantes, el nombre de sus aeropuertos (que es único) con la categoría de los mismos y el año de inauguración.

Así pues, los requerimientos de consulta y actualización más frecuentes son los siguientes:

- Lista de todos los vuelos existentes entre dos ciudades dadas.
- Lista de todos los vuelos existentes entre dos aeropuertos dados.
- Lista de la tripulación asignada a un vuelo en un día concreto.
- Lista de pasajeros de un vuelo en un día concreto.
- Listado de la clasificación en las finales de las pruebas.

## 4.15 Control de finanzas

El Ministerio de Hacienda ha decidido diseñar una base de datos para controlar las finanzas de los ciudadanos. La información que desea almacenar se detalla a continuación.

Se quiere tener información de los bancos que operan en el país, de cada uno se almacenará el código, que es único, el nombre y la dirección de la sede central. Los bancos tienen sucursales, al menos una, en distintos puntos geográficos. De cada sucursal interesa saber el número de la sucursal, que es único para cada banco, la dirección y el nombre del director.

De los ciudadanos del país interesa almacenar el número de la seguridad social, que es único, el D.N.I., que también es único, el nombre, la dirección y el teléfono.

Un ciudadano puede abrir una o varias cuentas en solitario o con otros ciudadanos. Cada cuenta se abre en una determinada sucursal y se identifica por un número de cuenta que es único en la sucursal; además, interesa saber el saldo actual de la cuenta, la fecha de apertura y el tipo de interés aplicado.

Otra información de interés hace referencia a los préstamos que ha otorgado cada sucursal a distintos ciudadanos. Cada préstamo, es otorgado por una sola sucursal y se identifica por un número de préstamo que es único en ella. Otras informaciones de interés relativas a los préstamos son: la fecha, el importe concedido, el tipo de interés aplicado y el ciudadano o ciudadanos a quienes se les ha concedido el préstamo. Por último, hay que saber en qué cuenta se cargan las anualidades del préstamo. Evidentemente, la cuenta debe ser de alguno de los ciudadanos que ha recibido el préstamo.

Por último, los bancos pueden realizar ofertas de distintos tipos de inversiones (fondos, plazos fijos, etc.) en los que los ciudadanos pueden invertir su dinero. Cada tipo de inversión tiene un código único, una descripción y un nombre. Un ciudadano puede realizar inversiones de los distintos tipos; cada inversión debe estar asociada a una de sus cuentas y se identifica por un código único y tiene una fecha y una cuantía.

## 4.16 Centro de acogida de perros

Se desea informatizar un centro de acogida de perros abandonados. Cada animal se registra con un número identificador, además se conserva de cada uno de ellos la fecha de llegada, una fecha aproximada de nacimiento (mes y año), raza, color del pelo, sexo, y en algunos casos una descripción referente a las circunstancias de su captura. También se almacena un nombre con el que se reconocerá al animal. El centro dispone de una serie de estancias o jaulas donde son asignados los animales durante su permanencia en el centro. Cada una de estas estancias tiene un código identificador, una capacidad máxima y la fecha de la próxima desinsectación.

Cuando los animales llegan al centro son reconocidos por un veterinario y si no se tiene constancia de que estén vacunados se les aplicarán las vacunas correspondientes a su edad y estado. De las diferentes vacunas que se administran en el centro se guarda el nombre que sirve para identificarla, a qué virus afecta, una descripción de los efectos secundarios, y un periodo de revacunación. Las vacunas se compran en envases con formato veterinario, cada uno lleva su número de lote, que es único por cada vacuna y una fecha de caducidad. Por cada una de las vacunas aplicadas habrá que guardar la fecha, la dosis que se utiliza y de qué lote se trataba.

Al centro llegan personas que quieren adoptar perros, de estas se almacena su nombre, su DNI, una dirección de contacto y necesariamente un teléfono, por lo menos. Estos benefactores se registran con un código interno.

Los perros del centro pueden estar en las estancias o adoptados. En este último caso puede serlo de manera temporal o permanente. Si el perro está en adopción temporal es necesario almacenar la fecha de la entrega, y la persona que lo adopta. Cuando un perro es adoptado de manera definitiva es necesario almacenar además el código del chip de identificación que se le inserta, y el nombre del veterinario que tendrá que castrar al animal. Solamente se desea saber el estado actual del animal, sin importar si se ha adoptado una o varias veces de manera temporal.

## 4.17 Una compañía eléctrica

Una compañía de distribución de energía eléctrica ha decidido desarrollar una base de datos para atender la gran cantidad de reclamaciones recibidas sobre los apagones de los últimos días.

La compañía tiene contratos con sus clientes. Cada contrato, que se identifica con un número y en el que consta la fecha de realización y la potencia contratada, es para el suministro de energía eléctrica a un único inmueble. Cada inmueble, que sólo puede tener un contrato, se codifica con la calle y el número dentro de cada población. De las poblaciones, que se identifican por su código, se almacena el nombre, la provincia y el número de habitantes. Además, en los contratos, que pueden ser para particulares o para empresas, figura los datos del cliente: el número de cliente, que los identifica, y la dirección de contacto. También se especifica, para el caso de contratos particulares, el DNI del cliente (que es único) y su nombre. Para los contratos con empresas, se detalla el NIF (que es único) y el nombre comercial.

Cuando se atiende una reclamación, se codifica y se hace constar la fecha en que se ha recibido y los datos del inmueble objeto de la reclamación. De cada inmueble sólo se puede atender una reclamación. Existen reclamaciones de dos tipos: ordinarias y extraordinarias. Las ordinarias corresponden a reclamaciones que se acogen a una serie de indemnizaciones que oferta la compañía para diferentes daños. Se dispone de una relación de estos daños, con su código, descripción e indemnización asociada. En consecuencia, en estas reclamaciones se detallan qué daños de los ofertados se reclaman.

Las reclamaciones extraordinarias no contemplan los daños ofertados por la empresa, sino que van acompañadas de una relación de los daños, numerada secuencialmente dentro de cada reclamación, con su correspondiente descripción e indemnización solicitada para cada uno de los daños reclamados.

La compañía dispone de información de las diferentes centrales distribuidoras de energía. Éstas tienen un código, un nombre y la potencia que es capaz de suministrar. Además, se conoce qué poblaciones abastece cada central (algunas poblaciones grandes son abastecidas por más de una central).

Finalmente, también se dispone, para cada central que abastece una población, de una relación de los intervalos horarios, por cada día, en los cuales se dejó de suministrar energía eléctrica a esa población.

## 4.18 Una cooperativa de vinos

La organización es una cooperativa que vende vinos embotellados de distintas regiones. Cada vino se distingue por el nombre, el año de producción y el tipo (hay vinos tintos, rosados y blancos), y se conoce también necesariamente la región de procedencia (identificada por un nombre) y el precio. Cada nombre de vino siempre tiene la misma procedencia. En cada región, los distintos años se obtienen producciones de calidad diferente (que se califica como mala, regular, buena, muy buena o excelente).

La cooperativa vende a socios, de los que sabe su nombre, dirección, ciudad, provincia y código postal, y a los que asigna un número de socio.

Además del vino, que es el producto principal, la cooperativa completa su oferta con productos de alimentación, que tienen cada uno un nombre diferente; además se indica la forma de presentación, el peso y el precio.

Todos los productos tienen un código que es único.

Cada socio puede hacer pedidos que pueden incluir cualquier combinación de productos, expresando la cantidad que desea de cada uno de ellos. Cada pedido tiene una fecha, un número de identificación, un importe total y un importe final (que se obtiene aplicando el descuento que proceda).

Hay una tabla de descuentos que contiene la cantidad de pesetas a partir de las cuales se aplica un porcentaje de reducción determinado al importe total de un pedido.

## 4.19 Elecciones a rector

Con el fin de organizar las elecciones a rector que se avecinan en la Universitat Politècnica de València,

se desea diseñar una base de datos cuyos requerimientos de información son los que siguen.

El censo está compuesto por las personas con vinculación a la universidad pertenecientes a los colectivos de profesores, alumnos y PAS.

De cada miembro del censo se debe conocer, el DNI que lo identifica, el nombre, el colectivo al que pertenece, así como el centro en el que realiza su actividad. Cada centro se identifica por un código y se debe saber de él, el nombre y los metros cuadrados de que dispone.

A las elecciones se presentan diferentes candidaturas, las cuales se identifican por un código único, se debe saber la fecha de presentación y el conjunto de miembros que la componen. Dichos miembros deben pertenecer al censo y se debe saber la posición que ocupan dentro de la candidatura. Un miembro del censo no puede aparecer dos veces ni en una ni en distintas candidaturas.

La votación se celebrará en diferentes mesas electorales; cada mesa se identifica por un número único dentro del centro en el que está, pudiendo haber varias mesas en un mismo centro.

Se debe saber en qué mesa de su centro le tocará votar a cada miembro del censo.

De las diferentes mesas electorales se deben conocer los cuatro miembros (pertenecientes al censo) que la componen, indicando además cuál de ellos es el presidente. Una persona sólo puede formar parte de una mesa.

De manera voluntaria, una candidatura puede proponer un interventor por mesa electoral. Dicho interventor ha de formar parte del censo y puede ser interventor de varias mesas. Un interventor no puede serlo para más de una candidatura.

## 4.20 Camino de Santiago

Se desea diseñar una base de datos sobre la Peregrinación a Santiago. Se quiere mantener información de los diferentes caminos (el camino francés, el camino aragonés, el camino vasco, etc.).

De cada camino se registra el nombre, que es único, el número de kilómetros totales, y una pequeña descripción. Cada camino se compone de un conjunto de etapas. Éstas se identifican por un número correlativo dentro de cada camino y se conoce los kilómetros de la misma y una serie de recomendaciones útiles para los peregrinos.

Cada etapa recorre un conjunto de localidades. Se debe conocer las distintas localidades por la que se pasa y en qué orden dentro de la etapa. Hay que tener en cuenta que por una localidad pueden pasar etapas de diferentes caminos, pero en un camino no se puede pasar dos veces por la misma localidad.

Las localidades se identifican por un código, tienen un nombre y se desea almacenar los sitios de interés que tienen. De cada sitio de interés, que se identifica con un número único dentro de cada localidad, se guarda el nombre y la descripción.

De las localidades que disponen de albergue se quiere saber el número de habitantes y qué albergues hay en la localidad. Los albergues se identifican por un código, tienen un nombre, una dirección y un precio.

Algunas localidades disponen de una oficina donde los peregrinos pueden pasar a cuñar el libro del peregrino que certifica que ha estado en esa localidad. De estas localidades se debe guardar el nombre de la oficina donde se realiza la certificación y la dirección completa de la misma. También se quiere tener un registro de los peregrinos que solicitan la certificación. De cada peregrino se guarda un identificador, su nombre completo y dirección y las localidades por las que ha pasado, indicando la fecha.

## 4.21 Locales de restauración

Se desea diseñar una base de datos para controlar los locales de restauración (bares y restaurantes) que permiten o no fumar, así como el registro de denuncias y sanciones a los mismos.

De cada local, se debe conocer el CIF, que es único, el nombre, la superficie en metros cuadrados, la dirección (calle, número y población), el teléfono (si lo tiene), el tipo de servicio (bar o restaurante).



Además, se deben registrar los datos del propietario, DNI, que lo identifica, nombre, dirección (calle, número y población) y teléfono. Se dispone de un registro de todas las poblaciones de España con la siguiente información: código de población, único para cada población, nombre, provincia y número de habitantes.

Los locales pueden ser de no fumadores, fumadores o mixtos. De los locales de no fumadores se debe conocer si disponen de terraza.

De los de fumadores, el tipo de sistema de extracción de humos que poseen, que debe ser de uno de los tipos que existen en el mercado, y de los cuales se conoce su código, nombre y características. De los locales mixtos, se debe saber las salas de que dispone, así como la superficie en metros cuadrados de cada una de ellas. Estas salas se identifican con un número único dentro de cada local.

Cuando se registra una denuncia sobre un local en el que no se puede fumar o en el que sólo se puede fumar en una zona, es decir, un local de no fumadores o mixto, se debe conocer, los datos del local denunciado, un código de denuncia único, la fecha de la denuncia, el motivo y el DNI del denunciante.

También se deben registrar las sanciones a los locales de no fumadores o mixtos.

Los tipos de sanción son los que contempla la ley y de los que se conoce su código, que es único, su descripción y la cuantía máxima y la mínima de la sanción.

Cada sanción se codifica con un número que es único dentro de cada local, se guarda la fecha, el tipo de sanción y el importe a pagar que deberá estar entre el importe máximo y mínimo del tipo de sanción. En los casos en los que la sanción sea consecuencia de una denuncia, se debe saber la denuncia que provocó la sanción (una denuncia sólo puede provocar una sanción).

## 4.22 Conflictos bélicos

Se desea diseñar una base de datos para almacenar información sobre los distintos conflictos bélicos (guerras) que han tenido lugar a lo largo de la historia y que ya han finalizado.

Cada guerra se codifica y tiene como información obligatoria el nombre, la fecha de inicio, los antecedentes históricos, la fecha de firma del tratado de paz y un resumen del resultado y de las consecuencias.

En una guerra se enfrentan dos bandos de los que se guarda un código, único para cada conflicto, y el nombre. Un país que ha combatido en una guerra lo ha hecho en uno de los dos bandos (supondremos que no se cambia de bando durante el conflicto). De cada país se quiere almacenar el nombre (que lo identifica), la bandera, la capital y el nombre del comandante en jefe de sus ejércitos durante cada conflicto indicando en qué periodo (puede haber más de uno, pero no a la vez). De cada persona que haya sido comandante de un país se guarda un código, el nombre, el año de nacimiento y de muerte (si es el caso).

Durante una guerra se libran batallas. Cada batalla tiene un código único en de la guerra en la que ocurrió, una fecha de inicio una fecha de fin y algunas tienen nombre. Hay que saber qué países han luchado en la batalla, que obviamente deben pertenecer a uno de los bandos enfrentados en el conflicto de la batalla. Hay que guardar el número de bajas de militares y de civiles de cada país en cada batalla en la que ha participado.

Las batallas pueden ser marítimas o terrestres. En el caso de las terrestres, cada batalla ha tenido lugar en una localidad concreta (no necesariamente de los países enfrentados). Cada localidad tiene un nombre único en el país al que pertenece hoy en día y hay que guardar su ubicación (latitud y longitud). Es interesante saber cuántos civiles de la localidad perecieron en la batalla.

Por último, se quiere almacenar información de distintos hechos especialmente relevantes que han ocurrido en los conflictos (p. e. el lanzamiento de la bomba atómica en la 2ª Guerra Mundial). Cada hecho histórico se ha codificado ha sucedido en un conflicto, tiene un nombre, una fecha, y una descripción, algunos hechos están asociado a un país.