

Introducción

Las bases de datos son un gran pilar de la programación actual, ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. En la actualidad se usan de forma mayoritaria las bases de datos relacionales, dominadas por distintos gestores de bases de datos a través de un lenguaje para la definición, manipulación y consulta de los datos, como puede ser SQL.

En la etapa de modelado o diseño conceptual se busca la manera de representar la estructura de una base de datos, en su mayoría relacional, buscando la representación de los esquemas.

Un modelo muy usado es el que plantearon los informáticos Peter Pin-Shan Chen y Edgar Frank Codd, y que llamaron **modelo entidad-relación**. Este método compuesto por diagramas se usa, entre otros fines, para diseñar los esquemas que permiten implementar bases de datos relacionales haciendo uso de aplicaciones informáticas denominadas **sistemas gestores de bases de datos**, representándolo a través de diagramas.

Estos diagramas ayudan a entender los datos y cómo se relacionan entre ellos, y deben ser complementados por un informe que liste los atributos y las relaciones entre cada elemento. Los elementos principales con los que cuenta este modelo son las entidades, los atributos simples, las relaciones, la cardinalidad entre las relaciones, los atributos que comportan la clave principal o primaria, los atributos que comportan la clave externa o foránea y otras estructuras más avanzadas que representan jerarquía, especialización y otros conceptos más específicos.

En esta unidad se procede al estudio de cada uno de dichos elementos usando ejemplos sencillos para alcanzar su entendimiento con el fin de llegar a obtener verdaderas estructuras complejas de representación del modelo conceptual. Al final de cada apartado se incluirá una serie de casos de uso que aglutina todos los elementos estudiados hasta el momento.

En el desarrollo de los contenidos se hace necesario adelantar algunos conceptos para analizar posibles mejoras y optimizaciones, pero que más tarde se detallan y se ejercitan para aprender estos, integrándolos con conceptos anteriores.

■ 2.1. El diseño conceptual en bases de datos

El diseño de bases de datos supone aquel proceso de generación de un diseño que permita dar soporte a la finalidad y a sus objetivos, con el fin de su implementación y explotación con el uso de sistemas gestores de bases de datos requeridos.

La intención de un modelado de datos es ofrecer unas técnicas para comprender el significado de los datos y facilitar los requisitos de información ubicados en un contexto, entendiendo como tal al universo limitado por el sistema de información que pretende gestionar toda esa información, por ejemplo, la gestión de una biblioteca, la gestión de las ventas

de una empresa, la de sus compras, la gestión de un campeonato de fútbol, la gestión del proceso de trazabilidad en entornos de producción, la gestión de la logística de distribución de una organización, etc. La construcción de un modelo conceptual en el diseño de bases de datos pretende dar forma sobre aquello de lo que se quiere guardar información, enfocándose en cuatro niveles principales:

1. **Detección de cada uno de los tipos de entidad:** detectar cada elemento del que se quiere almacenar información en el contexto dado.
2. **Detección de sus atributos:** se deben reconocer las propiedades o características de estas entidades. Los atributos comportan la forma abstracta o concreta de las entidades.
3. **Detección de las relaciones:** detectar la naturaleza de las relaciones que existen entre las entidades del contexto.
4. **Detección de las restricciones:** conocer las posibles restricciones de cada una de las propiedades de cada tipo de entidad, tanto para los atributos como para las relaciones.

El modelo de datos es dependiente de la visión que el diseñador tiene de los requisitos de información.

La educación de requisitos, también llamada **elicitación de requisitos** (del inglés *requirements eliciting*), es el conjunto de técnicas iniciales en la ingeniería de requisitos y busca conocer el dominio del problema, atendiendo al sistema actual, para identificar las necesidades de los que quieren explotar el sistema de información que automatizar. La comunicación con el cliente, parte fundamental que plantea el problema que resolver, es de vital importancia en estos procesos. El problema fundamental radica en la propia carencia del cliente en técnicas para informar de sus necesidades e incluso el desconocimiento real de estas.

Este complejo proceso nace con el análisis del dominio del problema, en el que se estudia la situación actual. En estas primeras acciones se desarrolla un glosario de términos y se obtiene el modelo de negocio actual. Cuando esta fase está lo suficientemente refinada, se detectan las diferentes necesidades que cubrir. Se obtienen documentos relativos a la información de las necesidades y el modelo del negocio que se quiere implantar. Las necesidades deben ser clasificadas, categorizadas y evaluadas en cuanto a prioridad, obteniendo el informe sobre prioridades.

El modelado del negocio actual describe cada uno de los procesos de negocio de la situación actual del negocio. Del mismo modo lo hace el modelado del negocio que se va a implantar. Se pueden usar diagramas de actividades y diagramas de casos de uso para representar de un modo global cada uno de esos procesos.

El análisis de las diferentes entrevistas llevadas a cabo con todo el personal competente de la organización a la que se le quiere mecanizar el proceso de negocio concluye con diferentes actas que especifican con claridad la información recopilada en cada una de las reuniones. Generalmente son los ingenieros de software, junto al equipo de analistas, los que realizan este tipo de trabajo.

Del proceso de **educción de requisitos**, que hace referencia a la parte de la información y su modelado, se obtiene una lista de **requisitos de información** que cumplir. De estos se va a partir para comenzar el proceso de diseño de las bases de datos. El conjunto de estos requisitos será la entrada del primer proceso del diseño de la base de datos: el modelo conceptual de datos.

El análisis de requisitos de información se relaciona con los aspectos referentes a los requisitos de un sistema en cuanto al subsistema de información. El conjunto de requisitos de un sistema completo se aborda en la ingeniería de requisitos, con el fin de determinar las necesidades de aquellos que quieren explotarlo, tanto los aspectos hardware como los aspectos software. Por extensión, los requisitos de información hacen referencia a las necesidades relacionadas con la información. Para ello, se trabaja bajo un modelo de datos que define las estructuras que son capaces de contenerlos y las condiciones para que represente el contexto real en el que se ubica. Los modelos de datos más conocidos son los conceptuales, los lógicos y los físicos. El primero hace referencia a la estructura y las condiciones de cumplimiento de los datos; el segundo se fundamenta en las operaciones más que en la descripción conceptual, y el tercero es el que da base a toda la construcción real haciendo uso de software que permite construir las bases de datos.

El modelo que mejor se adapta a la naturaleza de las bases de datos actuales que no son gigantes es el modelo relacional. Este está compuesto de elementos base denominados **relaciones**; tuplas que almacenan datos en una estructura bidimensional.

El modelo conceptual que mejor permite el diseño de las bases de datos de tipo relacionales es el **modelo entidad-relación**. Con este modelo se pretende obtener una representación simple sobre aquello de lo que se quiere conocer, sus características y las relaciones entre ellos. Estos elementos que dan base al diseño se denominan **entidades** y sus propiedades, **atributos** y **relaciones**.

Las bases de datos orientadas a objetos usan estructuras que no son tablas para almacenar la información. En la actualidad existen mecanismos para trabajar con bases de datos relacionales, bases de datos objetos y las orientadas a ambos usos, las bases de datos objeto-relacionales.

El diseño conceptual es la etapa inicial del modelado de datos y consiste en detectar sobre aquello de lo que se quiere almacenar información y cómo se relaciona.

■ ■ 2.1.1. Entidades

Las entidades representan personas y objetos que se diferencian entre sí, y de las que se desea almacenar información. Puede ser una entidad concreta como un ordenador, un libro, un profesor, un estudiante, un aula, un producto, etc., o una entidad abstracta con existencia conceptual, como puede ser una asignatura, una carrera, un departamento, una profesión, etcétera.

Teniendo en cuenta la abstracción de clasificación se distingue entre el tipo de entidades, que es la estructura genérica que describe un conjunto de entidades, y las entidades, que son los ejemplares de ese tipo de entidad. Es decir, el tipo de entidad es el resultado de la clasificación de un conjunto de entidades.

En realidad, en los diagramas entidad-relación se representa la colección de entidades o tipos de entidad que comparten los mismos atributos o propiedades. Por ejemplo, el tipo de entidad *profesor* aglutina a todas las instancias de dicho tipo de entidad que comparten los mismos atributos o características: *Antonio Marín Marea*, *Raquel Sánchez Mora*, *Iván Palacios Marín*, etc. Cada uno de estos profesores sería una entidad del tipo de entidad *profesor*. Comúnmente, se sintetiza esta dualidad reduciendo a entidad el tipo de entidad al que puede pertenecer un objeto o cosa, eso sí, siempre que compartan los mismos atributos.

Uno de los problemas en el diseño conceptual es la decisión si un determinado objeto o concepto se modela como un tipo de entidad o no. Puede ser un atributo, o formar un tipo en su clasificación, etc. Se proponen tres reglas generales que debe cumplir una entidad:

1. Una entidad tiene que tener existencia propia.
2. Cada ejemplar de un tipo de entidad, lo que denominamos **entidad**, debe poder distinguirse de las demás.
3. Todos los ejemplares de un tipo de entidad deben tener las mismas propiedades.

Ejemplos de entidades puede ser: profesor, estudiante, asignatura, producto, cliente, empleado, vendedor, comercial, departamento, transportista, vehículo, vuelo, vivienda, aula, hotel, sucursal, banco, etcétera.

La representación en el modelo entidad-relación para un tipo de entidad es un rectángulo en cuyo interior se incluye el nombre del tipo de entidad. A continuación, se representan los tipos de entidad: empleado, móviles, fabricante, cliente, proveedor, tienda, oficina, alquiler, marca y libro.



Figura 2.1. Los tipos de entidad indican los elementos de los que se quiere almacenar información. Se representa con un rectángulo.

En relación al tipo de entidad *MÓVILES*, en general, el nombre del tipo de entidad se escribe en singular, ya que identifica a un tipo de entidad: la entidad *Francisco Marín* es de tipo de entidad *Empleado*, siendo extraño decir la expresión «del tipo de entidad *Empleados*». En el ejemplo anterior, se ha escrito *MÓVILES* porque el diseñador hace referencia, que debería evitarse, a la tabla *Móviles*, que almacenará todos los registros del tipo *Móvil*.

2.1.2. Los atributos en los tipos de entidad

Los atributos definen o identifican las características de una entidad. Podemos decir que es el contenido de una entidad. Los atributos describen propiedades que posee cada miembro de un conjunto de entidades o tipo de entidad. Estos atributos pueden ser de diferentes tipos. Ejemplos de atributos para el tipo de entidad profesor puede ser: DNI, nombre, primer apellido, segundo apellido, número de la Seguridad Social, fecha de nacimiento, domicilio habitual, etc. Todas las entidades de dicho tipo de entidad comparten los mismos atributos. Las instancias de dichos atributos caracterizan de forma unívoca a dicha entidad, al menos, por uno de sus atributos.

Hay que tener en cuenta para la detección de los diferentes atributos de una entidad que, en ocasiones, un aparente atributo en realidad se trata de un tipo de entidad. No es difícil detectar este tipo de situaciones, pero debe prestarse atención a esta posibilidad. En general, se dice que una propiedad será un tipo de entidad si ella misma tiene sus propias propiedades. Se debe evitar usar tipos de entidad con tildes o con la eñe, ya que, en fases posteriores no se puede usar este tipo de caracteres en los identificadores de los objetos de la base de datos, por ejemplo, el tipo de entidad MÓVILES de la Figura 2.2.



Figura 2.2. Los atributos caracterizan a cada tipo de entidad y toman valores en cada instancia de un tipo de entidad, es decir, en cada entidad.

2.2. Las propiedades

Las propiedades de los tipos de entidad hacen referencia a sus características, es decir, sus atributos, pero también son las consecuencias inherentes de estar relacionados con otros tipos de entidades. Los atributos que determinan cómo es un tipo de entidad y que son compartidos por todas las instancias de ese tipo de entidad se pueden

clasificar atendiendo a sus especialidades. En una primera clasificación, estos son los tipos de atributos:

- Atributo principal o atributo secundario.
- Atributo simple o atributo compuesto.
- Atributo obligatorio u opcional.
- Atributo monovaluado o multivaluado.
- Atributo derivado.

■ ■ 2.2.1. Dominio de un atributo

El dominio de un atributo es el conjunto de valores permitidos para un atributo en particular. Para cada atributo, existe un dominio de este y hace referencia al tipo de datos con el que se almacenará o a restricciones en los valores que el atributo puede tomar. El dominio se puede indicar implícita o explícitamente. La primera a través de reglas; la segunda especificando cada uno de los valores que dicho atributo puede tomar. Puesto que un conjunto de entidades o tipo de entidad puede tener diferentes atributos, cada entidad se puede describir como un conjunto de pares de la forma (atributo, valor). Un ejemplo para el tipo de entidad profesor, su atributo DNI puede tomar 9 cifras numéricas. El atributo día de la semana en el que nació puede tomar uno de los valores del conjunto explícito Lunes, Martes, Miércoles, Jueves, Viernes, Sábado y Domingo. Ejemplos de pares serían (DNI, 44229539), (nombre, Antonio), (primer apellido, Postigo), (ciudad de nacimiento, Albany), etcétera.

■ ■ 2.2.2. Atributo principal y atributo secundario

De la colección de atributos que caracteriza a un tipo de entidad, uno o más de ellos, identifica unívocamente cada una de sus posibles ocurrencias. A este atributo se le denomina **atributo principal**, atributo clave, identificador principal o identificador candidato. Por ejemplo, el siguiente tipo de entidad llamado *Profesor* tiene los siguientes atributos: dni, nombre, PrimerApellido, SegundoApellido, dirección y numeroCuenta. De todos ellos, el atributo principal sería dni, ya que identifica cada profesor de tipo *Profesor*.

El atributo principal posee las siguientes características:

- Es único para cada ocurrencia de un tipo de entidad.
- Toda ocurrencia debe tener un valor.
- Este valor es el que se usará para relacionar los diferentes tipos de entidad.
- Todos los tipos de entidad deben tener un atributo principal.
- El atributo principal puede estar compuesto por varios atributos simples.

Su representación varía según el modelo. Haciendo uso del modelo E/R se usa una circunferencia rellena en negro (—● dni). En el modelo entidad-relación de este libro se usará la **letra P** delante del nombre del atributo y un punto · para separarlo (P-dni).

En la Figura 2.3 se puede observar la definición de cada uno de los atributos principales de cada tipo de entidad representados en la Figura 2.2.

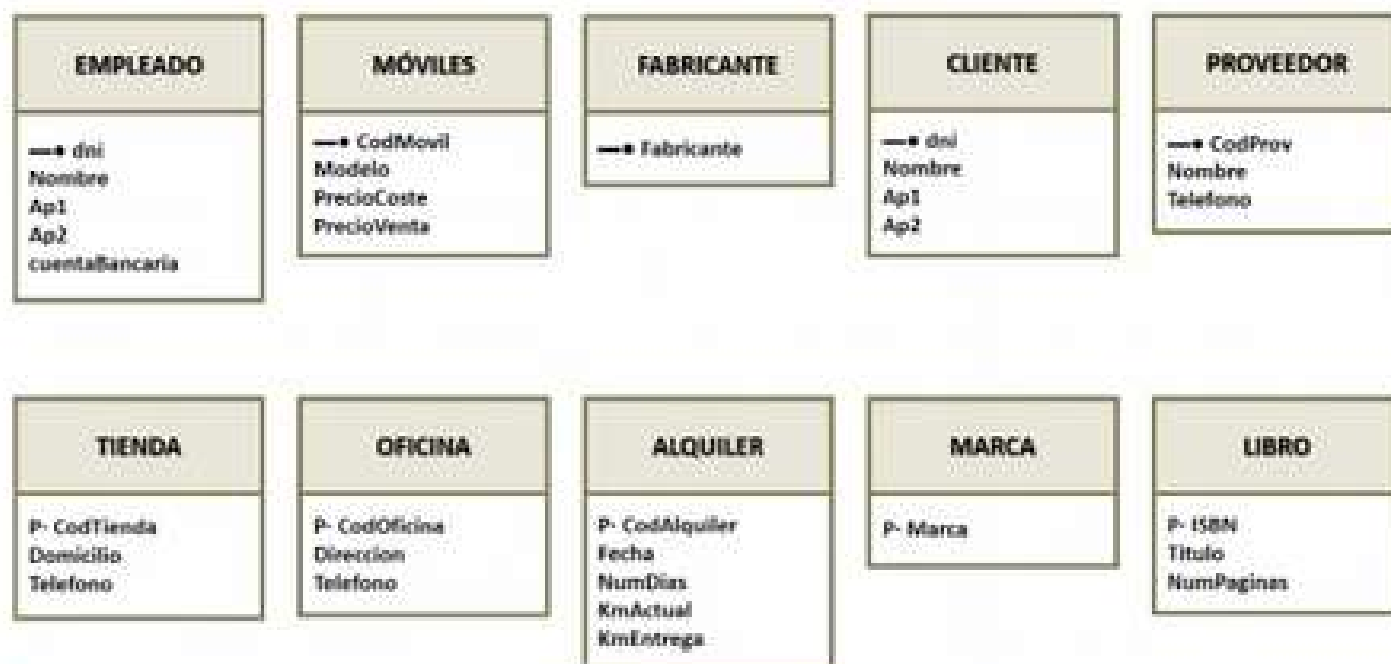


Figura 2.3. Todo tipo de entidad tendrá uno o más atributos que identifican unívocamente a cada una de sus instancias o entidades. Este se llama **atributo principal**.

Generalmente el atributo principal está compuesto por un solo atributo, pero puede ocurrir que haya más de un atributo simple. En este caso, se coloca el símbolo en cada uno de los atributos. A continuación, se representa un tipo de entidad con un atributo principal compuesto por varios atributos simples (Figura 2.4).

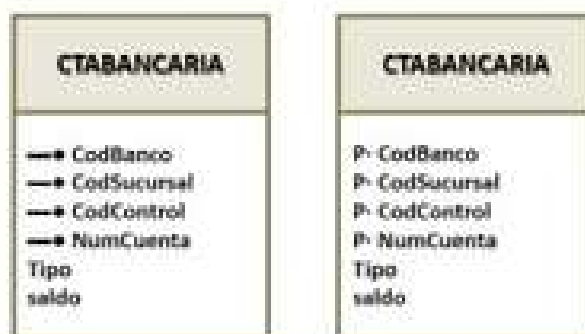


Figura 2.4. El atributo principal de un tipo de entidad puede estar compuesto por más de un campo.

El atributo principal de la cuenta bancaria está compuesto por los atributos simples código del banco (CodBanco), código de la sucursal (CodSucursal), código de control (CodControl) y el número de cuenta de diez dígitos dentro de dicha sucursal (NumCuenta).

Algunas entidades tienen **atributos secundarios**. El atributo secundario tiene las mismas características que el atributo principal, identifica unívocamente a cada una de las entidades y es obligatorio, pero no es el que se usa para relacionar los tipos de entidad. En el tipo de entidad *Profesor*, un atributo secundario podría ser el número de la Seguridad Social (NSS). Es único y obligatorio, pero no es el que se usa para relacionarlo, por ejemplo, con el tipo de entidad *Asignatura*.

Su representación varía según el modelo. Haciendo uso del modelo entidad-relación se usa una circunferencia más pequeña rellena en negro, —● dni. En el modelo de este libro se usará la **letra U** delante del nombre del atributo y un punto - para separarlo, U-NSS.

En el ejemplo de la Figura 2.5, se ha decidido que el atributo principal de *Empleado* sea Código del empleado, *CodEmpl*, y el secundario su *dni*. Del tipo de entidad *Cliente* se ha decidido *dni* como atributo principal y su número de Seguridad Social, *NSS*, como atributo secundario. Del tipo de entidad *Proveedor* se ha elegido *CodProv* como atributo principal y el código de identificación fiscal, *CIF*, como atributo secundario. Evidentemente, esto dependerá de los requisitos de información detectados por el analista o, también, por cuestiones que mejoren el diseño.



Figura 2.5. Un atributo también identifica a cada entidad, pero no es el atributo que se usa para relacionar los tipos de entidad, más bien se usa para las consultas de datos.

Los atributos se clasifican en:

- **Simples:** no contienen otros atributos.
- **Compuestos:** incluyen otros atributos simples.
- **Monovalorados:** solo toman un valor para la entidad.
- **Multivalorados:** pueden tomar varios valores para una misma entidad.
- **Derivados:** sus valores pueden conseguirse con operaciones sobre valores de otros atributos.
- **Nulos:** son aquellos que en el algún momento no existen o no se conoce su valor.

■ ■ 2.2.3. Atributos simples y compuestos

La definición dada de atributos hasta este momento se refiere a los atributos simples. Estos son aquellos que no pueden subdividirse en más atributos. Sin embargo, aquellos que pueden dividirse en partes se denominan **atributos compuestos**. Por ejemplo, para el atributo fecha de nacimiento puede existir la subdivisión día, mes y año de

nacimiento. El atributo cuenta bancaria, que consta de 20 dígitos numéricos, puede dividirse en banco, sucursal, dígitos de control y cuenta. El atributo domicilio puede dividirse en tipo de vía, nombre de la vía, número, planta, letra y resto del domicilio.

Los atributos compuestos son útiles para modelar situaciones en las que un usuario hace referencia al atributo compuesto como una unidad, pero otras se refieren a sus componentes. Si nunca se hace referencia a las partes de un atributo compuesto, no hay necesidad de subdividirlo. Para representarlo se usa una raya para cada parte del nombre compuesto. En la representación clásica los atributos, que son parte del atributo compuesto, se tabulan con respecto al nombre compuesto con el fin de indicar que son partes de él.



Figura 2.6. El atributo cuenta está compuesto por cuatro atributos denominados CodBanco, CodSucursal, CodControl, NumCuenta. Es por ello que se llama atributo compuesto.

2.2.4. Atributos multivalorados

Generalmente los atributos tienen un solo valor para cada entidad de un conjunto de entidades o tipo de entidad: los **atributos monovaluados**. Cuando un atributo puede tomar más de un valor para una misma entidad se denominan **atributos multivalorados**. Ya veremos que este tipo de atributos no cumplen las formas normales de Chen y Codd, y por lo tanto es necesario su transformación. A nivel conceptual, se puede hacer uso de los atributos multivalorados para simplificar en estos casos el modelo conceptual que se intenta representar. Ejemplos de atributos multivalorados pueden ser el atributo teléfonos, el atributo domicilios, el atributo idiomas que se habla, etcétera.

Para representarlo se usa una flecha delante del atributo. En la representación usada en este libro se utiliza la **letra M** delante del nombre del atributo. En la Figura 2.7 se representan algunos ejemplos usando ambas nomenclaturas.

2.2.5. Atributos con valores nulos

Se dice que un atributo toma un valor nulo cuando una entidad no tiene un valor para un atributo específico. Un valor nulo puede representar la no existencia del valor correspondiente para dicho atributo o puede representar que aún no se conoce, en el momento de la instancia u ocurrencia de esa entidad, su valor. Por ejemplo, al instanciar una entidad del tipo de entidad profesor, se desconoce el número de teléfono de contacto o no tiene segundo apellido al tratarse de un profesor nacido en un país que solo hace uso de uno.



Figura 2.7. Los atributos multivalorados permiten representar multiplicidades que evitan complicar la representación conceptual.



Figura 2.8. Los atributos no nulos son muy importantes, determinan la obligatoriedad de la instancia de un valor.

Para representarlo se usa un círculo sin relleno delante del atributo (—○). En la representación empleada en este libro se utiliza la **letra N** delante del nombre del atributo (o en la segunda representación el símbolo —○). En la Figura 2.8 se representan algunos ejemplos usando ambas nomenclaturas.

2.2.6. Atributos derivados

Los atributos derivados son aquellos que se pueden obtener realizando alguna operación con uno o más atributos de un tipo de entidad. En realidad, no es necesario almacenarlos a menos que dichas operaciones sean complejas y el almacenamiento de

dicho dato optimice temporalmente los accesos a estos evitando su cálculo continuo. En definitiva, su uso es cuestión de optimización espacial o temporal. Ejemplos de este tipo de atributos son la letra del DNI, el atributo edad si se dispone de la fecha de nacimiento, etc. Se representa a través del símbolo \square delante del nombre del atributo. En este libro se usa la **letra D**. Supóngase los siguientes requisitos de información:

- Se desea almacenar información de todos los campeonatos que una organización dirige. Se almacena su nombre (Descripcion), la fecha y hora en el que comienza el campeonato y la fecha y hora de fin, el número de días que ha durado (duración), y el número de partidas que se ha jugado durante el campeonato.
- De cada campeonato se registrará la hora de comienzo, duración en minutos y el resultado de cada una de las partidas jugadas en él.

Si en campeonato existen dos atributos para registrar la fecha y hora de comienzo y la de fin, a través de operaciones se puede calcular el número de días transcurridos. Por tanto, este atributo es derivado. Para el atributo número de partidas, *NumPartidas*, es fácil calcular dicho valor si se cuenta cuántas partidas se han relacionado en la relación *CampPartida* para dicho campeonato (esto último se entenderá mejor cuando se estudien las interrelaciones). Por tanto, este atributo también es un atributo derivado. La Figura 2.9 representa el resultado en el diseño conceptual usando las dos nomenclaturas (la primera es la que se ha empleado en este libro).



Figura 2.9. Los atributos derivados expresan valores que se conocen por otros medios, pero son útiles para acelerar los cálculos asociados.

Para definir las restricciones de los atributos se pueden utilizar diferentes nomenclaturas. En la Figura 2.10 se resumen estas pequeñas diferencias. La primera es la que sigue este libro.

ALUMNO	ALUMNO	ALUMNO
P•dni U•NSS Nombre Ap1 N•Ap2 MN•Telefono C•Localidad	P•dni (1,1) U•NSS (1,1) Nombre (1,1) Ap1 (1,1) N•Ap2 (0,1) MN•Telefono (0,n) C•Localidad (1,1)	—• dni —• NSS — Nombre — Ap1 —○ Ap2 →○ Telefono — Localidad

Figura 2.10. En la representación conceptual es muy normal indicar a cada atributo sus posibles restricciones. La nomenclatura que se usa en esta unidad es a través del uso de letras, la usada en la izquierda. Obviamente, como solo es una representación conceptual, se pueden usar otras nomenclaturas. La de la derecha es otra muy común.

Actividad resuelta 2.1.

Entidades y atributos. Caso de uso «Ventas supermercado versión 1.0»

A continuación, se lista un conjunto de requisitos de información. Detectar cuáles son los elementos de los que se quiere guardar información, los tipos de entidad y sus atributos, identificando los diferentes tipos.

- Requisito de información 1: se desea construir una base de datos para una compañía de supermercados, Supermercados NIRAM, que tiene supermercados por toda España. Se quiere almacenar la información de cada uno de ellos: dirección, metros cuadrados, si está en régimen de alquiler o de compra, y fecha en la que esta se efectuó. Se quiere saber quién es el director de cada supermercado. De los directores se almacena su DNI, nombre, apellidos, domicilio, teléfono y correo electrónico.
- Requisito de información 2: se registra información de los productos que se venden en los diferentes supermercados, con un código, descripción, familia y género al que pertenece, descuento para socios y precio e IVA.
- Requisito de información 3: cuando se realiza una venta, se quiere registrar la fecha, los productos que se venden, el vendedor que la realiza y, si procede, el cliente al que se le realiza. El cliente participa cuando se hace una venta a un socio que recibe descuentos especiales. Tanto de los clientes como de los vendedores se requiere almacenar su DNI, nombre, apellidos, domicilio, teléfono y correo electrónico.

Solución

El objetivo primero es localizar cada uno de esos elementos de los que se quiere guardar información, los tipos de entidad. Una vez detectados, es necesario conocer cuáles son sus características, es decir, sus atributos.

Al leer el requisito R1, se deduce que existen los tipos de entidad Supermercado y Director. Los atributos de Supermercado son CodSup, direccion, superficie, alquiler y fecha, donde el atributo principal es CodSup. Para Director los atributos son dni, nombre, prApellido, sgApellido, domicilio, telefono y email, donde dni se convierte en atributo principal.

Con la lectura del segundo requisito, R2, se deduce que existe el tipo de entidad Producto, con los atributos CodProd, descripcion, familia, genero, descuento, precio e iva.

Con la lectura del tercer requisito, R3, se deduce que existe el tipo de entidad Venta con los atributos CodVenta y fecha. Además, se deduce que existen los tipos de entidad Cliente y Vendedor con los atributos dni, nombre, prApellido, sgApellido, domicilio, telefono y email, donde dni se convierte en el atributo principal.

A continuación, se representa cada uno de los tipos de entidad con sus respectivos atributos. Se usa la representación con letras y la representación con símbolos. Cada tipo de entidad tiene su atributo principal (P, —●) y se indican aquellos que podrían no ser obligatorios (N, —○).



Figura 2.11. Representación de los tipos de entidad y atributos detectados en el caso de uso «Ventas supermercado versión 1.0».