

Modelo Entidad Relación

José Ramón Paramá Gabía

Capítulo 3

Modelo Entidad-Relación

Generalmente, el término *aplicación de base de datos* se refiere a una base de datos en particular (por ejemplo la base de datos BANCO que mantiene las cuentas de ahorro de sus clientes) y a los programas asociados, que implementan las consultas y actualizaciones de la base de datos (por ejemplo, programas que implementan actualizaciones de la base de datos correspondientes a los depósitos y reintegros de clientes). Por lo tanto, parte de la aplicación de base de datos requerirá el diseño, implementación y prueba de estos *programas de aplicación*, pero también requiere el diseño, implementación y prueba de la base de datos en sí misma. Tradicionalmente, se ha considerado que el diseño y prueba de los programas de aplicación pertenece más al dominio de la ingeniería del software que al de las bases de datos. Sin embargo, cada vez es más obvio que existe algo en común entre las metodologías de diseño de bases de datos y las de ingeniería del software. Es cierto que esas características comunes aumentarán, ya que las metodologías de diseño de base de datos tratan incluir conceptos de especificación de operaciones sobre objetos de base de datos, y que las metodologías de ingeniería del software especifican con más detalle la estructura de la base de datos.

Pero en este curso nos centraremos en las estructuras de bases de datos y en las restricciones durante el diseño de la base de datos.

3.1. Diseño de bases de datos

En el diseño de bases de datos se distinguen principalmente dos fases de diseño; la fase de modelado conceptual, que es la descripción del mundo real (una organización) de acuerdo con un modelo altamente semántico e independiente del SGBD en el que posteriormente se vaya hacer la implementación de la base de datos, y la fase de diseño lógico, en la cual se ha de obtener un esquema que responda a la estructura lógica específica del SGBD que se vaya utilizar en cada caso, por lo que dicho esquema está sometido a las restricciones que imponga el modelo del SGBD en concreto.

La Figura 3.1 representa la forma de llegar desde la parcela del mundo real que se está analizando a la base de datos física. En un primer paso, con la ayuda del modelo conceptual, se obtiene el esquema conceptual. A continuación, aplicando al esquema conceptual las reglas del modelo de datos propio del SGBD que se va utilizar, se obtiene el esquema lógico; de éste se pasa al esquema interno, donde el objetivo es conseguir la máxima eficiencia de cara a la máquina y al problema específico. Por último se implementa la base de datos específica en los soportes secundarios. La estructura física se ha de rellenar con los

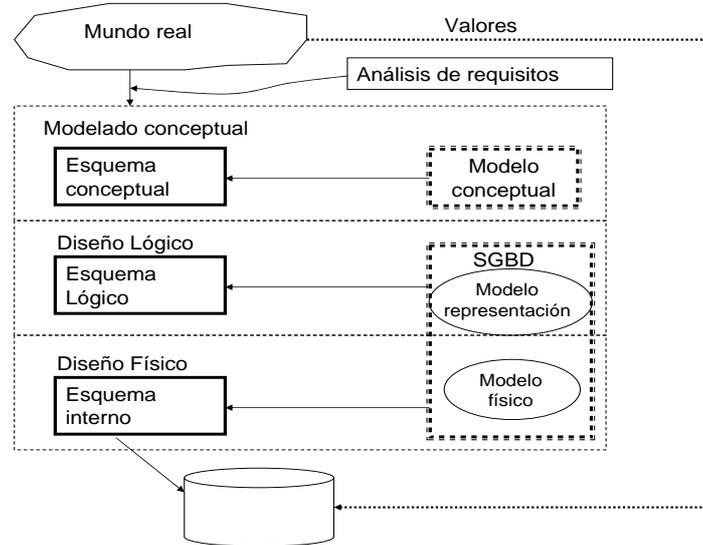


Figura 3.1: Proceso de diseño de BDs.

valores que se obtienen por observación de los sucesos del mundo real.

De este modo, el proceso de diseño de una base de datos puede ser dividido en cuatro pasos:

1. **Análisis de requisitos**, el primer paso es entender qué datos van a ser almacenados en la base de datos, qué aplicaciones deben ser construidas sobre ella y qué aplicaciones son más frecuentes y por lo tanto requieren un cuidado especial para obtener un rendimiento adecuado.

Esta fase es un proceso que incluye discusiones con los grupos de usuarios, estudio de los actuales sistemas operacionales y cómo se espera que cambien, examen de cualquier información disponible sobre las aplicaciones existentes que van a ser sustituidas o completadas por la aplicación de base de datos que se va diseñar, etc. Existen multitud de metodologías para realizar este paso, incluso existen herramientas que lo automatizan.

2. **Diseño Conceptual**, la información recogida en el paso anterior es usada para desarrollar el esquema conceptual. Este paso se suele realizar con el modelo Entidad-Relación.
3. **Diseño Lógico**, una vez elegido el SGBD a utilizar, se convierte el diseño conceptual de la base de datos en un esquema de base de datos del SGBD elegido. Este esquema se suele llamar, como apuntamos anteriormente, el esquema lógico.
4. **Diseño Físico**, en este paso se deben considerar las cargas de trabajo que debe soportar la BD que se está diseñando para refinarla de modo que cumpla con los requisitos de rendimiento especificados. Este paso puede implicar cosas tan simples como construir índices sobre algunos ficheros, o puede implicar un rediseño importante del esquema obtenido de los pasos anteriores.

Una vez terminado el proceso de diseño, seguramente se requiera sucesivas *puestas a punto* (*tunning* en inglés), donde los pasos anteriores se intercalan y se repiten hasta que el diseño es satisfactorio.

Los conceptos y técnicas que incluye un SGBD son claramente útiles para alguien que desea implementar o mantener las interioridades de un sistema de bases de datos. Sin embargo, es importante reconocer que usuarios responsables, diseñadores de bases de datos y ABDs deben conocer también cómo funciona el SGBD. Un buen conocimiento de las interioridades del SGBD es esencial para aquellos usuarios que deseen aprovechar al máximo el SGBD y realizar buenos diseños de bases de datos. Por eso, los Capítulos 1 y 2 son fundamentales en la formación de un experto en SGBD, a pesar de que justamente parte de los objetivos de estos sistemas es ocultar a los usuarios muchos de estos conceptos.

3.2. Ejemplo guía

Durante este capítulo vamos utilizar el ejemplo de una empresa para ilustrar los conceptos del modelo Entidad-Relación (ER). Supongamos que después del análisis de requisitos se obtiene la siguiente especificación:

- La empresa está organizada en departamentos. Cada departamento tiene un nombre único, un número único y siempre tiene un empleado que lo dirige. Nos interesa la fecha en que dicho empleado comenzó a dirigir el departamento. Un departamento puede estar distribuido en varios lugares.
- Cada departamento controla un cierto número de proyectos, cada uno de los cuales tiene un nombre y número únicos, y se efectúa en un solo lugar. Un departamento puede no estar involucrado en proyectos.
- Almacenaremos el nombre, número de seguridad social, dirección, salario, sexo y fecha de nacimiento de cada empleado. Todo empleado está asignado a un departamento, pero puede trabajar en varios proyectos, que no necesariamente estarán controlados por el mismo departamento. Nos interesa el número de horas por semana que un empleado trabaja en cada proyecto, y también quién es supervisor directo de cada empleado. No todo empleado es supervisor.
- Queremos mantenernos al tanto de los familiares de cada empleado para administrar sus seguros. De cada familiar almacenaremos el nombre, sexo, fecha de nacimiento y parentesco con el empleado.

3.3. Tipos de entidad, conjuntos de entidad, atributos y claves

El modelo ER fue introducido en el año 1976 por P. Chen. El modelo ER describe los datos como entidades, relaciones y atributos.

3.3.1. Entidades y atributos

Una *entidad* es una “cosa” u “objeto” en el mundo real que es distinguible de todos los demás objetos. Una entidad puede ser un objeto con existencia física (una persona, un ordenador, un hotel o un empleado) o un objeto conceptual (una empresa, una enfermedad,

un viaje). Cada entidad tiene propiedades específicas, llamadas *atributos*, que la describen. Por ejemplo, una entidad empleado puede describirse por su nombre, edad, dirección, salario y puesto de trabajo. Una entidad particular tendrá un *valor* para cada uno de sus atributos. Los valores de los atributos que describen a cada entidad constituyen una parte decisiva de los datos almacenados en la base de datos.

Supongamos que tenemos dos entidades e_1 y e_2 . La entidad e_1 es una entidad *empleado* con atributos: Nombre, Dirección, Edad y Puesto de trabajo, y valores para esos atributos: “Pedro”, “Calle Real Nº1 2ºA, 15002, A Coruña”, 25 y “Conserje” respectivamente. La entidad e_2 es una entidad departamento, con atributos “Número de departamento”, “Nombre de departamento” y “Localidad”, con valores: 10, “Ventas” y “A Coruña” respectivamente.

En el modelo ER se manejan varios tipos distintos de atributos: *simples* o *compuestos*; *monovaluados* o *multivaluados* y *almacenados* o *derivados*.

Atributos simples y compuestos

Los *atributos compuestos* se pueden dividir en componentes más pequeños, que representan atributos más básicos con su propio significado independiente. Por ejemplo, el atributo Dirección de la entidad empleado se puede dividir en Calle, Número, Piso, Puerta, Código Postal y Ciudad, así para la entidad e_1 los valores serían: “Real”, 1, 2, “A”, 15002 y “A Coruña” respectivamente. Los atributos no divisibles se denominan atributos *simples* o *atómicos*. El valor del atributo compuesto es la concatenación de los valores de los atributos simples que los constituyen.

Los atributos compuestos son útiles para modelar situaciones en las que un usuario en unas ocasiones hace referencia al atributo compuesto como una unidad, pero otras veces se refiere específicamente a sus componentes. Si sólo se hace referencia al atributo compuesto como un todo, no hay necesidad de subdividirlo en sus atributos componentes.

Atributos monovaluados y multivaluados

En su mayoría, los atributos tienen un solo valor para una entidad particular; estos atributos se denominan de *monovaluados*. Por ejemplo, Edad es un atributo monovaluado de Empleado. Pero hay casos en que un atributo puede tener varios valores para una entidad concreta, por ejemplo un atributo Hijos para un Empleado, evidentemente puede haber empleados con más de un hijo. Este tipo de atributos se denominan *multivaluados*.

Atributos almacenados y derivados

Consideremos por ejemplo el atributo Edad de los empleados. No tiene sentido almacenar dicho atributo, puesto que al poco tiempo de almacenar en la base de datos la edad de un empleado, ésta puede quedar desfasada. Lo más lógico sería almacenar la fecha de nacimiento, y cada vez que se consultara la edad de un empleado, un procedimiento la calculara a partir de la fecha de nacimiento del empleado y la fecha en la que se realiza la consulta. En este caso, el atributo Fecha de Nacimiento es el *atributo almacenado* y Edad sería el *atributo derivado*.

3.3.2. Tipos de entidad, conjuntos de entidades

Por lo regular, una base de datos contiene grupos de entidades similares. Por ejemplo, una empresa que da empleo a cientos de empleados seguramente querrá almacenar información

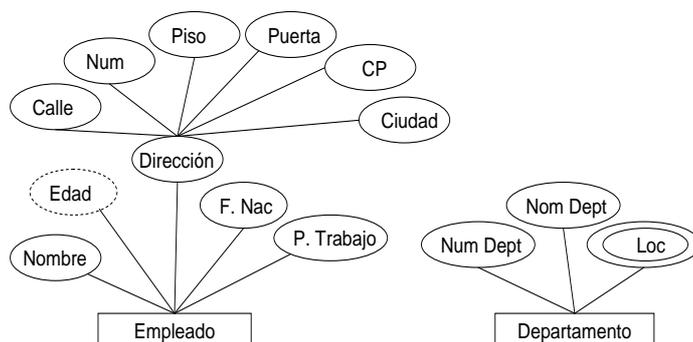


Figura 3.2: Diagrama ER de los tipos de entidad Empleado y Departamento.

similar sobre cada uno de ellos. Estas entidades empleado comparten los mismos atributos, pero cada entidad tiene *su propio valor* (o valores) para cada atributo. Un *tipo de entidad* define una colección (o conjunto) de entidades que poseen los mismos atributos. Cada tipo de entidad de la base de datos se describe por su nombre y sus atributos. En el siguiente ejemplo, tenemos dos tipos de entidades: *Empleados* y *Departamentos*, cada una con sus atributos.

Nombre del tipo	Empleado				Departamento		
Atributos	Nombre	Dirección	Edad	P. Trabajo	Num Dept	Nom dept	Loc

Cada ocurrencia de los tipos de entidad anteriores tendrá valores para cada uno de los atributos, como vimos para las entidades e_1 (ocurrencia del tipo de entidad *Empleado*) y e_2 (ocurrencia del tipo de entidad *Departamento*). La colección de todas las entidades de un tipo particular de entidad en la base de datos en cualquier instante de tiempo se llama *conjunto de entidades*; al conjunto de entidades se le suele dar el mismo nombre que al tipo de entidad. Por ejemplo, EMPLEADO se refiere tanto al tipo de entidad como al conjunto del entidades de todos los empleados de la base de datos.

Un tipo de entidad se representa en los diagramas ER¹ como un rectángulo que encierra el nombre del tipo de entidad. Los nombres de atributos se encierran en óvalos y se conectan con su tipo de entidad mediante líneas rectas. Los atributos compuestos se conectan con sus atributos componentes mediante líneas rectas. Los atributos multivaluados aparecen en óvalos de doble contorno.

En la Figura 3.2 podemos observar la representación de los tipos de entidad Empleado y Departamento. El atributo *Edad* es un atributo derivado de *Fecha de Nacimiento*, *Dirección* es un atributo compuesto y *Localidad* es un atributo multivaluado, indicando que un departamento puede tener sedes en distintas localidades.

3.3.3. Claves

Es necesario tener una forma de especificar cómo las entidades dentro de un conjunto de entidades dado son distinguibles. Conceptualmente las entidades individuales son distintas;

¹Se va utilizar una notación próxima a la propuesta originalmente por Chen. Desafortunadamente, hay muchas variaciones.

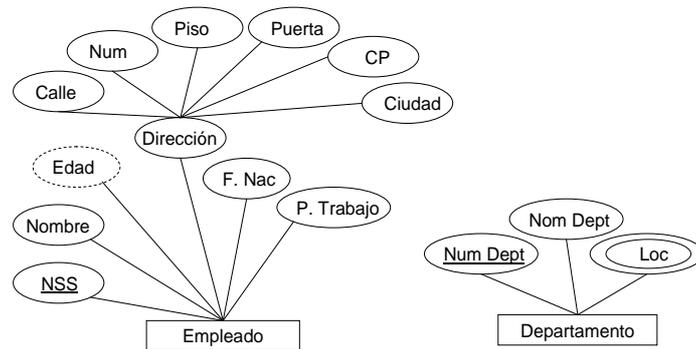


Figura 3.3: Diagrama ER con atributos clave.

desde una perspectiva de bases de datos, sin embargo, la diferencia entre ellas se debe expresar en término de sus atributos.

Por lo tanto, los valores de los atributos de una entidad deben ser tales que permitan *identificar unívocamente* a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos.

Una *clave* permite identificar un conjunto de atributos suficiente para distinguir las entidades entre sí.

Una **superclave** es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades. Por ejemplo, el atributo Número de Seguridad Social (NSS) es una superclave del conjunto de entidades Empleado de la Figura 3.3, porque no hay dos empleados con el mismo NSS. Análogamente, la combinación de los atributos Nombre y NSS es una superclave del conjunto de entidades Empleado. El atributo Nombre de Empleado no es una superclave, porque varias personas pueden tener el mismo nombre.

El concepto de superclave no es suficiente para lo que aquí se propone, ya que, como se ha visto, una superclave puede contener atributos innecesarios. Si K es una superclave, entonces también lo es cualquier superconjunto de K . A menudo interesan las superclaves tales que los subconjuntos propios de ellas no son superclave. Tales superclaves mínimas se llaman **claves candidatas**.

Es posible que conjuntos distintos de atributos pudieran servir como clave candidata. Supóngase que una combinación de Nombre y Dirección de los Empleados es suficiente para distinguir entre los miembros del conjunto de entidades Empleado. Entonces, los conjuntos $\{NSS\}$ y $\{Nombre, Dirección\}$ son claves candidatas. Aunque los atributos NSS y $Nombre$ juntos puedan distinguir entidades Empleado, su combinación no forma una clave candidata, ya que el atributo NSS por sí solo es una clave candidata.

Se usará el término **clave primaria** para denotar una clave candidata que es elegida por el diseñador de la base de datos como el elemento principal para identificar las entidades dentro de un conjunto de entidades. Una clave (primaria, candidata y superclave) es una propiedad del conjunto de entidades, más que de las entidades individuales.

La clave primaria se debería elegir de manera que sus atributos nunca, o muy raramente, cambien. Por ejemplo, el campo dirección de un Empleado no debería formar parte de la clave primaria, porque probablemente cambiará.

En la notación de los diagramas ER, los atributos de un conjunto de entidades que son miembros de la clave primaria aparecen subrayados dentro del óvalo, como se ilustra en la Figura 3.3.

3.3.4. Dominios

Cada uno de los atributos simples de un tipo de entidad está asociado a un conjunto de valores (o *dominio* de valores), que especifica los valores que es posible asignar a ese atributo para cada entidad individual. Podemos suponer que los valores del atributo *Puesto de Trabajo* del tipo de entidad *Empleado* puede únicamente tomar el conjunto de valores (dominio) {*Conserje, Vendedor, Ingeniero Junior, Ingeniero Senior*}. Se podría especificar dominios menos concretos, por ejemplo, el dominio del atributo Nombre podría ser el conjunto de las cadenas de caracteres alfabéticos separadas por caracteres de espacio en blanco. Los dominios no se representan en los diagramas ER.

3.3.5. Diseño conceptual inicial de la base de datos Empresa

Ahora podemos definir los tipos de entidad de la base de datos EMPRESA, que se basa en los requisitos de la Sección 3.2. Este es un primer paso, más adelante iremos añadiendo más elementos a nuestro diseño.

En el caso que nos ocupa, podemos encontrar cuatro tipos de entidad:

1. Un tipo de entidad DEPARTAMENTO con los atributos Nombre, Número, Localizaciones, Gerente y Fecha-Incio-Gerente. Localizaciones es el único atributo multivaluado. Tanto Nombre como Número son claves candidatas de este tipo de entidad.
2. Un tipo de entidad PROYECTO con los atributos Nombre, Número, Localización y Departamento-Controlador. Tanto Nombre como Número son claves candidatas.
3. Un tipo de entidad EMPLEADO con los atributos Nombre, NSS, Sexo, Dirección, Salario, Fecha-Nacimiento, Departamento y Supervisor. Tanto Nombre como Dirección pueden ser atributos compuestos; sin embargo, esto no se especificó en los requisitos. Debemos remitirnos a los usuarios para ver si alguno de ellos va hacer referencia a los componentes individuales de Nombre (Nombre de pila, Apellido1, Apellido2) o de Dirección.
4. Un tipo de entidad FAMILIAR con los atributos Empleado, Nombre-Familiar, Sexo, Fecha-Nacimiento y Parentesco (con el empleado).

Hasta aquí no hemos representado el hecho de que un empleado puede trabajar en varios proyectos, ni el número de horas a la semana que un empleado trabaja en cada proyecto. Tenemos dos alternativas: un atributo compuesto multivaluado de EMPLEADO llamado Trabaja-en, con componentes simples (Proyecto, Horas), la otra alternativa es añadir al tipo de entidad PROYECTO un atributo compuesto multivaluado llamado Trabajadores, con componentes individuales (Empleado, Horas). En la Figura 3.4 elegimos la primera alternativa. El atributo Nombre del tipo de entidad EMPLEADO aparece como atributo compuesto, probablemente como resultado de haber consultado con los usuarios.

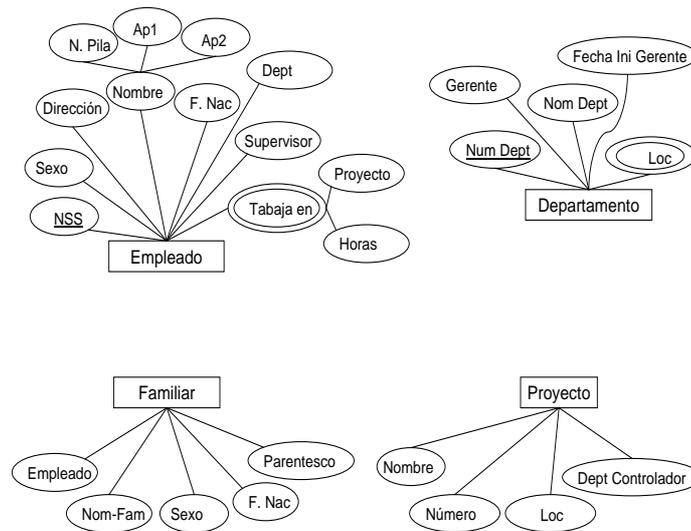


Figura 3.4: Diagrama ER con tipos entidad y atributos.

3.4. Relaciones, tipos de relación, roles y restricciones estructurales

En la Figura 3.4 hay varias *relaciones* implícitas entre los diversos tipos de entidad. De hecho, siempre que un atributo de un tipo de entidad hace referencia a otro tipo de entidad, hay alguna relación.

Por ejemplo, el atributo Gerente de DEPARTAMENTO se refiere a un empleado que dirige el departamento; el atributo Departamento Controlador de PROYECTO se refiere al departamento que controla el proyecto; el atributo Supervisor de EMPLEADO se refiere a otro empleado (el que supervisa a ese empleado); el atributo Departamento de EMPLEADO se refiere al departamento para el cual trabaja el empleado, etc. En el modelo ER, estas referencias no se deben representar como atributos, sino como **relaciones**.

3.4.1. Tipos, conjuntos e instancias de relaciones

Un *tipo de relación* R entre n tipos de entidad E_1, E_2, \dots, E_n define un conjunto de asociaciones, o *conjunto de relaciones*, entre entidades de estos tipos. Al igual que en los tipos de entidad y conjuntos de entidades, un tipo de relación y su correspondiente conjunto de relaciones se designan normalmente con el mismo nombre R . En términos matemáticos, R es un conjunto de *instancias de relación* r_i , donde cada r_i asocia n individuales (e_1, e_2, \dots, e_n) y cada entidad e_j de r_i es miembro del tipo de entidad E_j , $1 \leq j \leq n$. Por tanto, un tipo de relación es una *relación matemática* sobre E_1, E_2, \dots, E_n , que también puede definirse como un subconjunto del producto cartesiano $E_1 \times E_2 \times \dots \times E_n$. Se dice que cada uno de los tipos de entidad E_1, E_2, \dots, E_n *participa* en el tipo de relación R y, de manera similar, que cada una de las entidades individuales e_1, e_2, \dots, e_n participa en la instancia de la relación $r_i = (e_1, e_2, \dots, e_n)$.

En términos informales, cada instancia de relación r_i de R es una asociación de entidades, donde la asociación incluye exactamente una entidad de cada tipo de entidad participante.

3.4. RELACIONES, TIPOS DE RELACIÓN, ROLES Y RESTRICCIONES ESTRUCTURALES

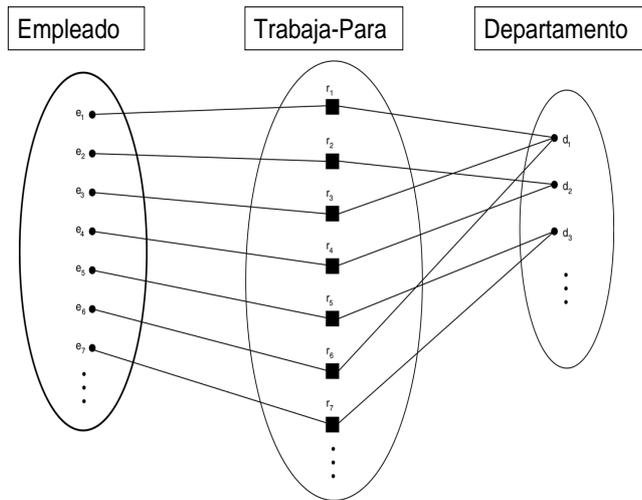


Figura 3.5: Instancias de la relación Trabaja-Para entre Empleado y Departamento.

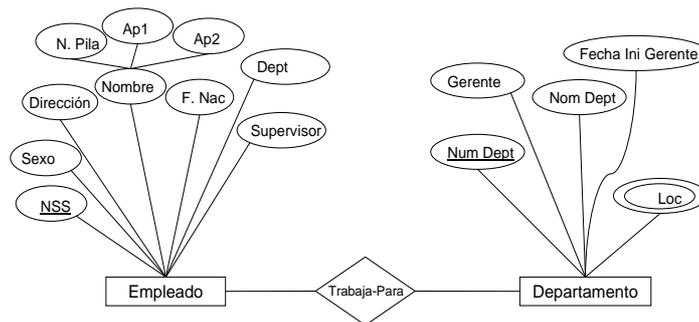


Figura 3.6: El tipo de relación TRABAJA-PARA.

Cada una de estas instancias de relación r_i representa el hecho de que las entidades que participan en r_i están relacionadas entre sí de alguna manera. Por ejemplo, consideremos un tipo de relación TRABAJA-PARA entre los tipos de entidad EMPLEADO y DEPARTAMENTO, que asocia a cada empleado con el departamento para el que trabaja. Cada instancia de relación de TRABAJA-PARA asocia una entidad empleado y una entidad departamento. En la Figura 3.5 se ilustra el ejemplo, donde cada instancia de relación r_i aparece conectada a las entidades departamento y empleado que participan en r_i . En la situación presentada por la Figura 3.5, los empleados e_1, e_3 y e_6 trabajan en el departamento d_1 ; e_2 y e_4 trabajan en d_2 , y e_5 y e_7 trabajan en d_3 .

En los diagramas ER, los tipos de relación se representan como rombos conectados mediante líneas rectas con los rectángulos que representan a los tipos de entidad participantes. El nombre de tipo de relación (si es necesario) aparece dentro del rombo, como se puede apreciar en la Figura 3.6.

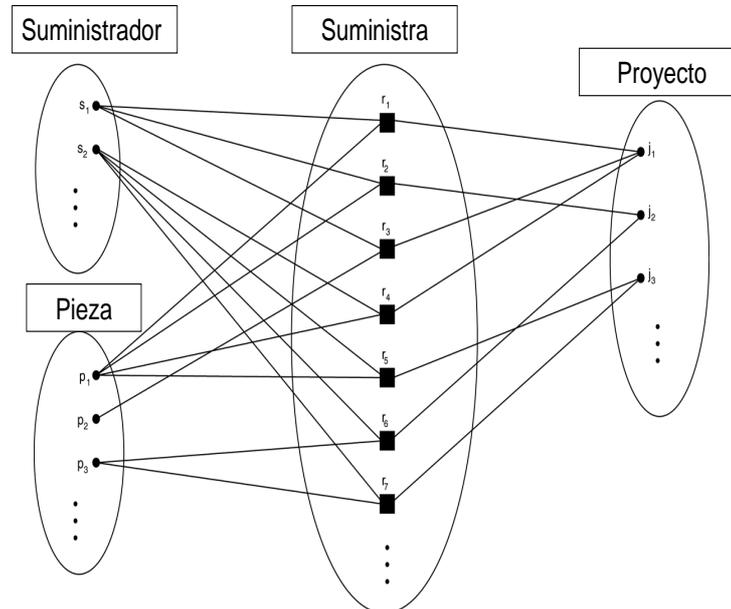


Figura 3.7: Instancias de un tipo de relación ternario.

3.4.2. Grado de relación, nombres de rol y relaciones recursivas

El *grado* de un tipo de relación es el número de tipos de entidad que participan en él. Así, el tipo de relación TRABAJA-PARA es de grado dos. Los tipos de relación de grado dos se llaman binarios, y los de grado tres se llaman ternarios.

En la Figura 3.7 se muestra un ejemplo de tipo de relación ternario, SUMINISTRAR, donde cada instancia de relación r_i asocia tres entidades (un proveedor s_k , una pieza p_j y un proyecto j_l) siempre que s_k suministre la pieza p_j al proyecto j_l . Los tipos de relación pueden tener cualquier grado, pero las más comunes son las binarias.

En ocasiones resulta conveniente considerar un tipo de relación en términos de atributos, como vimos en la Sección 3.3.5. Tomemos como ejemplo el tipo de relación TRABAJA-PARA de la Figura 3.6. Consideremos el atributo llamado *Departamento* (*Dept* en la figura) del tipo de entidad EMPLEADO, cuyo valor para cada entidad empleado es (una referencia) a la entidad departamento a la cual pertenece el empleado. Por tanto el dominio de este atributo es el conjunto de *todas las entidades* DEPARTAMENTO. Este atributo Departamento, implementa el tipo de relación TRABAJA-PARA.

Cada tipo de entidad que participa en un tipo de relación desempeña un *rol* o *papel* específico en el tipo de relación. El *nombre de rol* indica el papel que una entidad participante del tipo de entidad desempeña en cada instancia del tipo de relación, y ayuda a explicar el significado de la relación. Por ejemplo, en el tipo de relación TRABAJA-PARA, EMPLEADO desempeña el papel de *empleado* o *trabajador* y DEPARTAMENTO tiene el papel de *departamento* o *patrón*.

Los nombres de papeles no son técnicamente necesarios en los tipos de relación en los que todos los tipos de entidad participantes son distintos, ya que cada nombre del tipo de entidad se puede usar como el nombre del papel. Sin embargo, en algunos casos el mismo tipo de entidad participa más de una vez en un tipo de relación con diferentes papeles. En

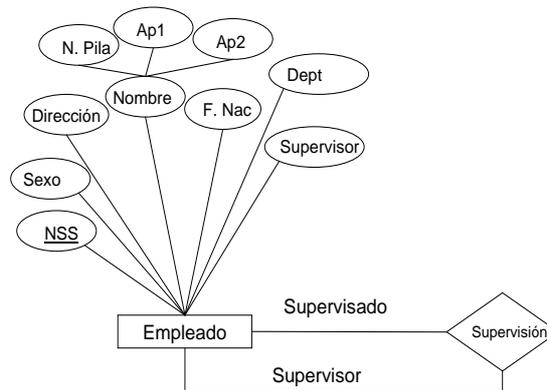


Figura 3.8: Relación recursiva.

tales casos el nombre del papel resulta indispensable para distinguir el significado de cada participación. Estos tipos de relación se denominan *relaciones recursivas*, y la Figura 3.8 muestra un ejemplo.

El tipo de entidad EMPLEADO participa dos veces en SUPERVISIÓN: una vez con el papel de supervisor (jefe), y una vez en el papel de supervisado (o subordinado). Cada instancia de relación r_i en SUPERVISIÓN asocia dos entidades empleado e_j y e_k , una de las cuales desempeña el papel de supervisor y la otra el de supervisado.

3.4.3. Restricciones sobre los tipos de relación

Los tipos de relación suelen tener ciertas restricciones que limitan las posibles combinaciones de entidades que pueden participar en los correspondientes tipos de relación. Estas restricciones se determinan a partir de la realidad, es decir del significado que tienen los tipos de entidad y tipos de relación en el mundo real, y no depende de los conjuntos de entidades o conjuntos de relación que en un momento dado se puedan estar considerando o almacenando en la base de datos. Por ejemplo, podríamos tener una empresa en donde los empleados sólo pueden trabajar para un departamento.

Estas restricciones se representan en el diagrama ER. Podemos distinguir dos tipos principales de restricciones asociadas a tipos de relación: *cardinalidad* y *participación*.

Cardinalidad

La *cardinalidad* de un tipo de relación binaria especifica el número de instancias de relación en los que puede participar una entidad. Por ejemplo, en el tipo de relación binaria TRABAJA-PARA, DEPARTAMENTO:EMPLEADO tiene una cardinalidad 1:N, lo que significa que cada departamento puede estar relacionado con muchos empleados, pero un empleado sólo puede estar relacionado con (trabajar para) un departamento (en la Figura 3.5, se puede observar un ejemplo de entidades e instancias de relación). Las cardinalidades pueden ser N:M, 1:N y N:1. Hay un caso especial de cardinalidad 1:N que es 1:1.

Un ejemplo de tipo de relación 1:1 es DIRIGE (Figura 3.9, que relaciona una entidad departamento con el empleado que dirige este departamento (como un empleado sólo puede trabajar para un departamento, por lo tanto la relación es 1:1)

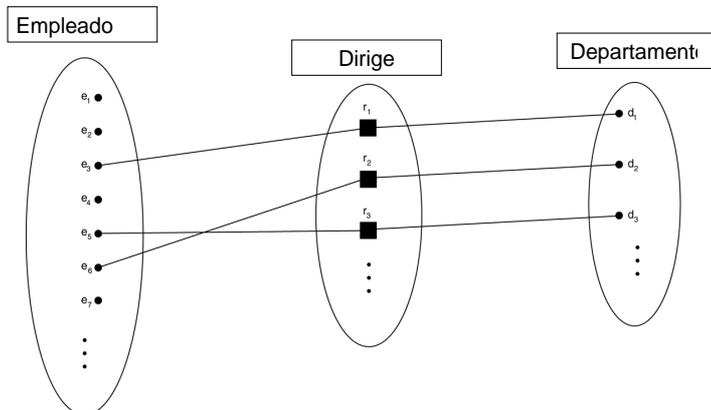


Figura 3.9: Relación 1:1.

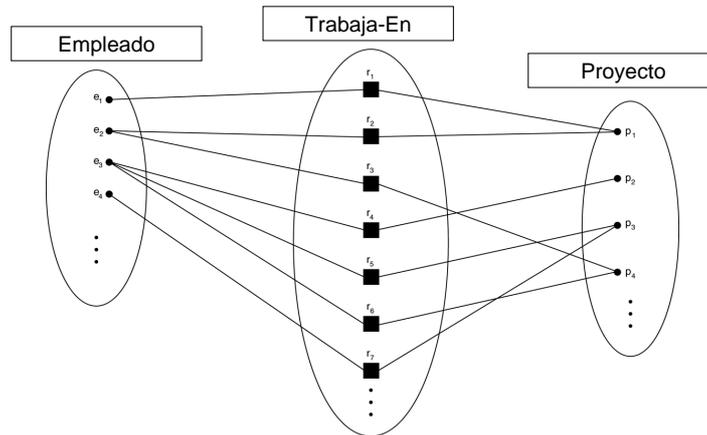


Figura 3.10: Relación N:M.

Recordamos al lector, de que a pesar de ilustrar el ejemplo con conjuntos de entidades y relaciones, la cardinalidad no depende del contenido de los conjuntos de entidades y relaciones en un momento dado, sino del significado en el mundo real de los *tipos* de entidad y relación. El tipo de relación TRABAJA-EN (Figura 3.10) tiene cardinalidad N:M, porque un empleado puede trabajar en varios proyectos, y en un proyecto pueden trabajar varios empleados.

Restricciones de participación y dependencia de existencia

La restricción de *participación* especifica si la existencia de una entidad depende de que esté relacionada con otra entidad a través del tipo de relación. Hay dos tipos de restricciones de participación, *total* y *parcial*, que ilustramos con un ejemplo. Si la política de una empresa establece que *todo* empleado debe pertenecer a un departamento, un empleado sólo puede existir si participa en una instancia del tipo de relación TRABAJA-PARA (Figura 3.5). Se dice que la participación de EMPLEADO en TRABAJA-PARA es una *Participación Total*, porque toda entidad del conjunto de entidades empleado debe estar relacionada con (al menos) una

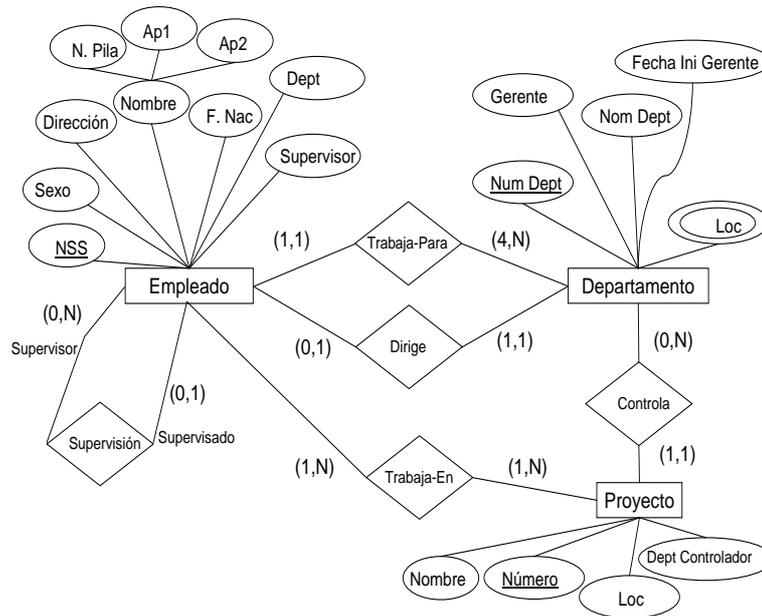


Figura 3.11: Diagrama ER con notación de mínimos máximos.

entidad departamento a través de TRABAJA-PARA. La participación total recibe a veces el nombre de *dependencia de existencia*. En la Figura 3.9 no cabe esperar que todo empleado dirija un departamento, así que la participación de EMPLEADO en el tipo de relación DIRIGE es *parcial*, lo que significa que algunas (parte del conjunto de) entidades empleado están relacionadas con una entidad departamento a través de DIRIGE, pero no necesariamente todas.

En los diagramas ER representaremos la cardinalidad y la restricción de participación con una de las múltiples posibilidades, la **notación de mínimos-máximos**. Se asocia un par de números enteros (mín, máx) a cada *participación* de un tipo de entidad E en un tipo de relación R , donde $0 \leq \text{mín} \leq \text{máx}$ y $\text{máx} \geq 1$. Los números significan que, para cada entidad e de E , e debe participar en al menos mín y como máximo en máx instancias del tipo de relación R en todo momento. En esta notación, $\text{mín}=0$ indica participación parcial, mientras que $\text{mín} > 0$ implica participación total. Los valores de máx correspondientes a las participaciones de los tipos de entidad en el tipo de relación nos indican la cardinalidad. En la Figura 3.11, se puede ver un diagrama ER con la notación mínimos máximos.

3.4.4. Atributos de los tipos de relación

Los tipos de relación también pueden tener atributos, similares a los tipos de entidad. Por ejemplo, para registrar el número de horas por semana que un empleado trabaja en un proyecto, podemos incluir un atributo Horas para el tipo de relación TRABAJA-EN de la Figura 3.11. Otro ejemplo sería incluir la Fecha-Inicio-Gerente en el tipo de relación DIRIGE.

En el caso del atributo Fecha-Inicio-Gerente, no hay mucha diferencia en incluir el atributo tal y como está en la Figura 3.11 (asociado al tipo de entidad DEPARTAMENTO) o asociado al tipo de relación DIRIGE, puesto que es una relación 1:1, y por lo tanto como podemos ver en la Figura 3.9, a cada entidad Departamento, le corresponde una única entidad del tipo

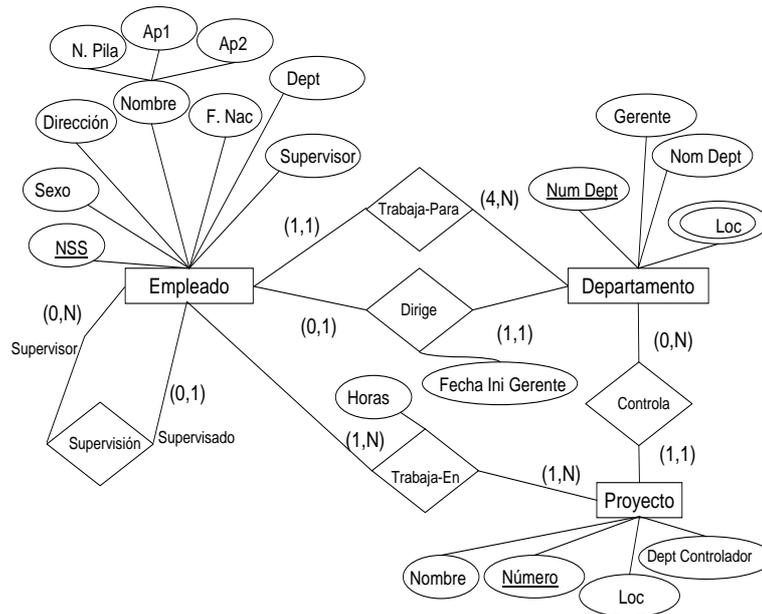


Figura 3.12: Diagrama ER con atributos en los tipos de relación.

de relación DIRIGE. Sin embargo, en el caso del atributo Horas, no es posible asignarlo al tipo de entidad PROYECTO (ni a EMPLEADO) porque tal y como podemos observar en la Figura 3.10, a cada entidad proyecto, le puede corresponder más de una instancia del tipo de relación, es decir, varios empleados. El atributo Horas sólo puede estar asignado a un par (entidad proyecto e_p , entidad empleado e_j) indicando las horas que un empleado trabajó para un proyecto, así que la única solución es que sea un atributo del tipo de relación.

Resumiendo, los atributos de los tipos de relación 1:N y 1:1 se pueden trasladar a al menos uno de los tipos de entidad participantes. En el caso de 1:1 a cualquiera de los dos tipos de entidad participantes, mientras que en el caso 1:N se puede trasladar al tipo de entidad que tiene una participación con cardinalidad 1. Como vimos en el ejemplo anterior, el atributo Fecha-Inicio-Gerente (al ser una relación 1:1) se puede colocar en cualquiera de los dos tipos de entidad. Sin embargo, si quisiésemos saber para cada empleado la fecha desde la cual trabajan para el departamento al que están asignados, ese atributo se podría colocar en el tipo de relación TRABAJA-PARA o bien en el tipo de entidad EMPLEADO (el que tiene cardinalidad 1 en la relación).

En el caso de los tipos de relación N:M, algunos atributos pueden estar determinados por la *combinación de la entidades participantes* en una instancia del tipo de relación, y no por alguna de ellas sola. Como vimos con el ejemplo del atributo Horas, tales atributos *deberán especificarse como atributos del tipo de relación*. Así el ejemplo de la Figura 3.11, después de añadir el atributo Horas y recolocar el atributo Fecha-Inicio-Gerente, queda tal y como se muestra en la Figura 3.12.

3.4.5. Tipos de entidad débil

Los tipos de entidad que no tienen atributos clave propios se denominan *tipos de entidad débiles*. En contraste, *los tipos de entidad regulares* que tienen atributo/s clave se suelen

llamar *tipos de entidad fuertes*. Las entidades que pertenecen a un tipo de entidad débil se identifican por su relación con entidades específicas de otro tipo de entidad, en combinación con algunos de los valores de sus atributos. Decimos que este otro tipo de entidad es el *tipo de entidad propietario* o *tipo de entidad padre*, y llamamos al tipo de relación que las une, *relación identificador* del tipo de entidad débil. Un tipo de entidad débil siempre tiene una *restricción de participación total* (dependencia de existencia) con respecto a su relación identificador, porque una entidad débil no se puede identificar sin una entidad propietaria. Sin embargo no toda dependencia de existencia da lugar a un tipo de entidad débil. Por ejemplo, una entidad EXPEDIENTE DE UN ALUMNO no puede existir sin que exista una entidad ALUMNO, aunque puede tener un NÚMERO DE EXPEDIENTE que lo identifique, y por tanto no es una entidad débil.

Consideremos el tipo de entidad FAMILIAR, relacionado con EMPLEADO, que sirve para llevar el control de los familiares de cada empleado a través de un tipo de relación 1:N. Los atributos de FAMILIAR son Nombre (nombre de pila del familiar), Fecha de nacimiento, Sexo y Parentesco (con el empleado).

Es posible que dos familiares de dos empleados distintos tengan los mismos valores de Nombre, Fecha de nacimiento, Sexo y Parentesco, pero seguirán siendo entidades distintas. Se identificarán como entidades distintas sólo después de determinar la entidad *empleado particular* con la que está relacionada cada una de ellas.

Normalmente, los tipos de entidad débiles tienen una *clave parcial*, que es el conjunto de atributos que pueden identificar de manera única las entidades débiles relacionadas con la misma entidad propietaria. En nuestro ejemplo, si suponemos que nunca dos familiares del mismo empleado tendrán el mismo nombre, el atributo Nombre de FAMILIAR será la clave parcial. En el peor de los casos, todos los atributos de la entidad débil formarán la clave parcial.

La clave primaria *real* de los tipos de entidad débil estará formada por la clave parcial más la clave primaria del tipo de entidad propietario.

En los diagramas ER, un tipo de entidad débil y su relación identificador se distinguen rodeando el rectángulo y el rombo con líneas dobles (tal y como se muestra en la Figura 3.13). Los atributos de la clave parcial se subrayan con una línea punteada o discontinua.

3.4.6. Eliminando atributos que implementan relaciones

En el diagrama de la Figura 3.4 y posteriores, se incluyen atributos asociados a tipos de entidad que representan relaciones. Por ejemplo, en el tipo de entidad EMPLEADO tiene el atributo DEPARTAMENTO (“Dept” en la Figura) que representa el departamento para el cual trabaja el empleado. Esa relación, se representa en el diagrama ER de la Figura 3.13 con el tipo de relación TRABAJA-PARA. En los diagramas ER no se incluyen los atributos que representan relaciones, se entiende que con el tipo de relación ya se representa el vínculo existente².

Con respecto a la Figura 3.4, lo mismo ocurrirá con el atributo Gerente del tipo de entidad DEPARTAMENTO (se elimina dado que se representa el vínculo con el tipo de relación DIRIGE), el atributo Fecha-Inicio-Gerente también desaparece de DEPARTAMENTO, ya que lo habíamos desplazado al tipo de relación en la Figura 3.12. En el tipo de entidad PROYECTO se elimina el atributo Departamento-Controlador (Dept Controlador en la Figura) que ya es

²Exclusivamente, a efectos didácticos, durante este curso se mantendrán indicándolo con una marca.

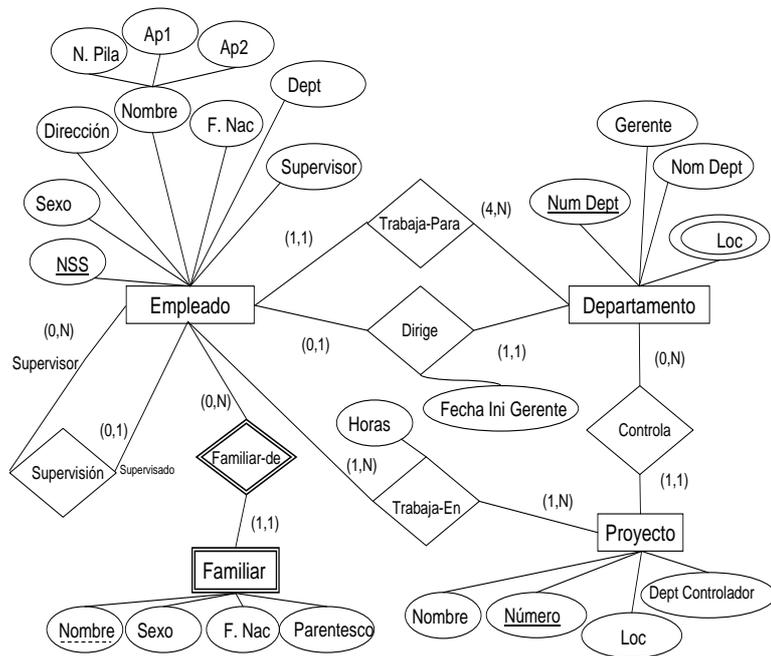


Figura 3.13: Diagrama ER con un tipo de entidad débil.

representado por el tipo de relación CONTROLA. Del tipo de entidad EMPLEADO, como ya mencionamos desaparece el atributo Departamento (por el tipo de relación TRABAJA-PARA), el atributo Supervisor, implementado por el tipo de relación SUPERVISIÓN y finalmente el atributo multivaluado compuesto Trabaja-en que es implementado por el tipo de relación del mismo nombre. El resultado final se puede ver en la Figura 3.14.

3.4.7. Notación alternativa

Existen multitud de alternativas a la hora de la notación del modelo ER. La variante más común es similar a la presentada hasta el momento pero con pequeñas diferencias a la hora de representar la cardinalidad y la participación.

En la Figura 3.15, vemos como la cardinalidad se expresa también añadiendo un 1 o una N en la línea que une el tipo de entidad con el tipo de relación. Note que la ubicación es la contraria a la notación mínimos máximos, tal y como se muestra en la Figura 3.16. La participación total se indica con una doble línea que conecta el tipo de entidad (con participación total) con el rombo correspondiente, mientras que la participación parcial se representa con una línea simple.

3.5. Elecciones de diseño para el diseño conceptual ER

En ocasiones es difícil decidir cuándo un concepto particular del problema a tratar debe ser modelizado como un tipo de entidad, un atributo, o un tipo de relación. En esta sección, damos algunas breves pautas para saber qué elementos elegir en determinadas situaciones.

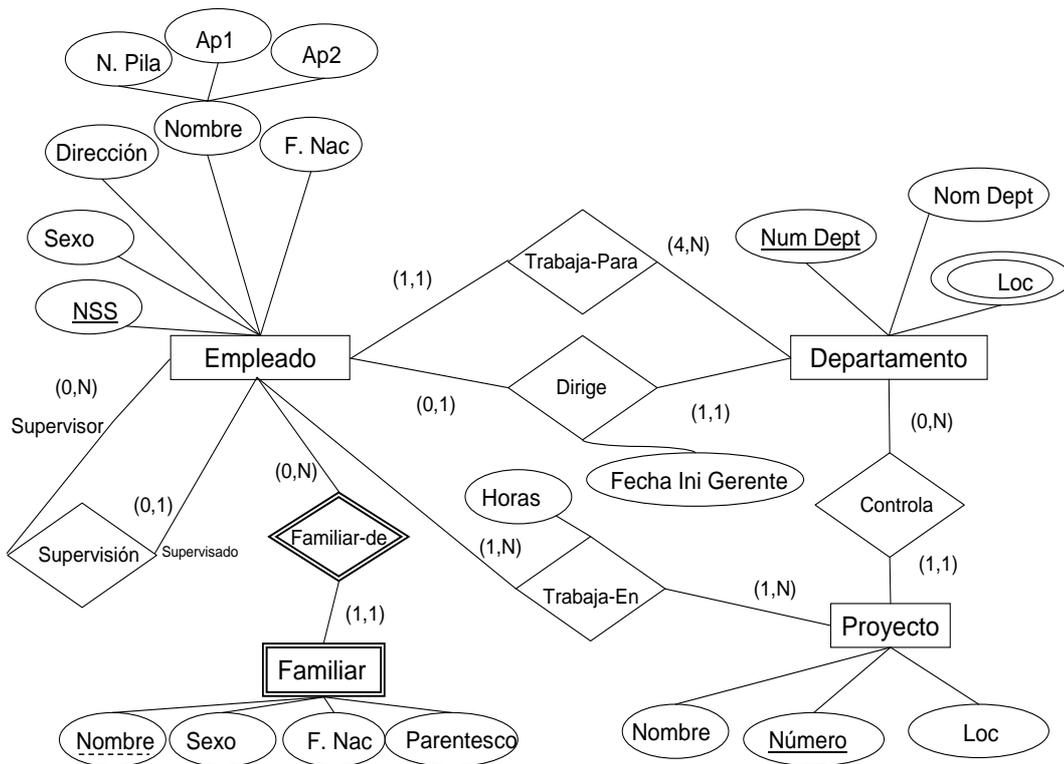


Figura 3.14: Diagrama ER final.

Uso de tipos de entidad o atributos

Considérese el tipo de entidad EMPLEADO (dentro de una universidad) con los atributos Nombre, Cargo (de gestión, por ejemplo, director de departamento, director de centro, vicerrector, etc.) y Complemento Retributivo (asociado al cargo). Se puede argumentar fácilmente que los Cargos dan lugar a un tipo de entidad por sí mismo con atributos: Denominación y Complemento-Retributivo. Si se toma este punto de vista, el tipo de entidades EMPLEADO debe ser redefinido como sigue:

1. El tipo de entidades EMPLEADO con el atributo Nombre.
2. El tipo de entidades CARGO con atributos: Denominación y Complemento-Retributivo.
3. El tipo de relación EMPLEADO-CARGO, que denota la asociación entre empleados y los cargos que desempeñan (puede que un empleado tenga más de un cargo).

¿Cuál es, entonces, la diferencia principal entre esas dos definiciones de un empleado? Al tratar los cargos como un atributo, implica que cada empleado tiene a lo sumo un único cargo. Al tratar cargo como un tipo de entidad CARGO permite que los empleados puedan tener varios cargos (incluido ninguno). Sin embargo, se podría definir fácilmente Cargo (y Complemento-Retributivo) como un atributo multivaluado para permitir varias cargos por empleado. La diferencia principal es que al tratar los cargos como un tipo de entidad, otros tipos de entidad pueden tener relaciones con el tipo de entidad CARGO. Por ejemplo, podríamos tener material

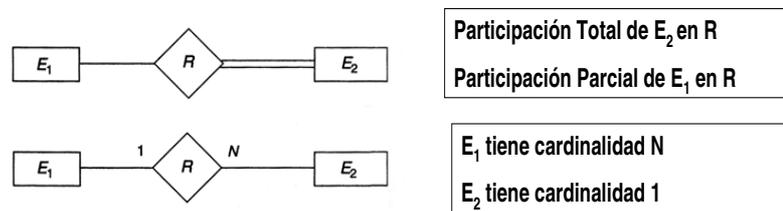


Figura 3.15: Notación alternativa.

auxiliar (por ejemplo coches) asignados a determinados cargos, es decir, tendríamos otro tipo de relación ASIGNADO entre los tipos de entidad CARGO y (por ejemplo) el tipo de entidad AUTOMÓVIL, indicando que cada entidad automóvil está asignada a un cargo en particular (y no a la persona que puede ser destituida en cualquier momento).

Por lo tanto, dado que el cargo tiene relaciones independientes del empleado al que está asignado, es apropiado modelarlo como un tipo de entidad separado del empleado. De todos modos, no es sencillo dar un método específico para definir qué es un atributo y qué es un tipo de relación. Las distinciones dependen de cada caso en particular, analizando la semántica del atributo en cuestión y las posibles relaciones con otros tipos de entidad.

Un error común es usar la clave primaria de un tipo de entidad como atributo en otro tipo de entidad, en lugar de usar un tipo de relación. Es decir, por ejemplo, dejar el atributo Departamento en el tipo de entidad Empleado tal y como está en la Figura 3.4, en lugar de incluir el tipo de relación TRABAJA-PARA que implementa la asociación entre empleados y los departamentos para los que trabajan.

Uso de conjuntos de entidades o conjuntos de relaciones

No siempre está claro si es mejor expresar un objeto mediante un tipo de entidad o con un tipo de relación. En la Figura 3.14 se asumió que un proyecto se modelaba como un tipo de entidad. Una alternativa es modelar un proyecto no como un tipo de entidad, sino como un tipo de relación entre empleados y departamentos, con los atributos Nombre, Número, Localidad y Horas como atributos del tipo de relación. Cada proyecto se representa mediante una instancia de ese tipo de relación.

Si cada proyecto está asociado exactamente con un empleado y con un departamento,

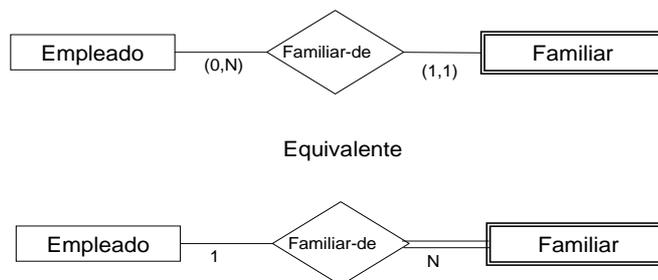


Figura 3.16: Equivalencia de las dos alternativas.

se puede encontrar satisfactorio el diseño en el que los proyectos se representan como tipos de relación. Sin embargo, con este diseño no se puede representar la situación (más común) de que un proyecto involucre más de un empleado y/o departamento. Habría que definir una instancia de relación por cada empleado y/o departamento en un proyecto común. Por ejemplo, si en un proyecto están implicados dos empleados (e_i y e_j) y dos departamentos (d_k y d_l), habría que definir cuatro instancias del tipo de relación PROYECTO ($\{e_i, d_k\}$, $\{e_i, d_l\}$, $\{e_j, d_k\}$, $\{e_j, d_l\}$) cada una de ellas replicando los atributos Nombre, Número y Localización (del proyecto) y todas las correspondientes a cada uno de los dos empleados replicando el atributo Horas. Es decir, cada una de las cuatro instancias correspondientes al proyecto que estamos considerando, repite los mismos valores de los atributos Nombre, Número y Localización, las instancias $\{e_i, d_k\}$, $\{e_i, d_l\}$ repiten el valor del atributo Horas para e_i y $\{e_j, d_k\}$, $\{e_j, d_l\}$ repiten el valor del atributo Horas para e_j .

Surgen dos problemas como resultado de esta réplica: 1) los datos se almacenan varias veces, desperdiciando espacio de almacenamiento; y 2) las actualizaciones pueden dejar potencialmente los datos en un estado inconsistente, al diferir los valores de los atributos en dos (o más) instancias del tipo de relación PROYECTO (debido a una actualización en una instancia y no en otras) que representan al mismo proyecto que, por tanto, deberían tener los mismos valores.

Este problema no se presenta en la Figura 3.14 dado que PROYECTO se modela como un tipo de entidad.

3.6. Problemas con los modelos ER

En esta sección examinaremos un tipo de problemas que pueden aparecer cuando se está creando un diagrama ER. Estos problemas se llaman las *trampas de conexión*, y normalmente ocurren debido a una mala interpretación del significado de ciertas relaciones. Examinaremos los dos tipos principales de trampas de conexión, llamadas la *trampa del abanico* y la *trampa del sumidero*, e ilustraremos cómo identificar y resolver estos problemas en diagramas ER.

En general, para identificar las trampas de conexión, debemos asegurarnos de que el significado de un tipo de relación está completamente entendido y claramente definido. Si no entendemos las relaciones, podríamos crear un modelo que no es una representación fidedigna del “mundo real”.

3.6.1. La trampa del abanico

La trampa del abanico ocurre cuando un modelo ER representa una relación entre tipos de entidad, pero el camino entre algunas entidades es ambiguo.

Una trampa de abanico puede aparecer si dos o más relaciones 1:N salen del mismo tipo de entidad. En la Figura 3.17 se presenta un caso potencial de trampa de conexión. Se presentan dos tipos de relación 1:N, (TRABAJA-PARA y TRABAJA-EN), que emanan del mismo tipo de entidad (DEPARTAMENTO).

Este modelo representa el hecho de que un Departamento puede trabajar en *una o más* secciones de la empresa y tiene *uno o más* empleados. Sin embargo, aparece un problema cuando deseamos saber qué empleados trabajan en una Sección particular. Para apreciar el problema, examinamos algunas instancias de los tipos de relación TRABAJA-PARA y TRABAJA-EN, tal y como se muestra en la Figura 3.18.

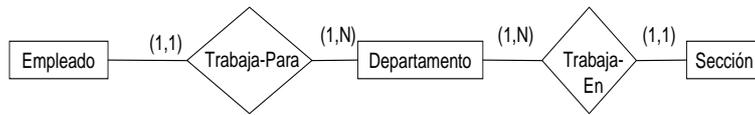


Figura 3.17: Ejemplo de trampa de de abanico.

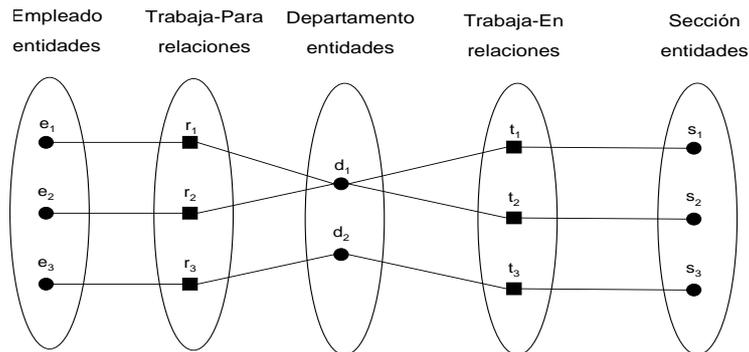


Figura 3.18: Ejemplo de trampa de de abanico utilizando conjuntos de entidades e instancias de tipos de relación.

Si formulamos la consulta: "¿en que Sección trabaja el empleado e_1 ?", somos incapaces de dar una respuesta concreta atendiendo a la Figura 3.18. Sólo podemos decir que e_1 trabaja en la sección s_1 o en la s_2 . La imposibilidad de responder esta pregunta es el resultado de la trampa del abanico asociada a la mala interpretación de las relaciones entre EMPLEADO, DEPARTAMENTO y SECCIÓN. Podemos resolver el problema reestructurando el modelo ER original para representar la asociación correcta entre esas entidades, tal y como se muestra en la Figura 3.19.

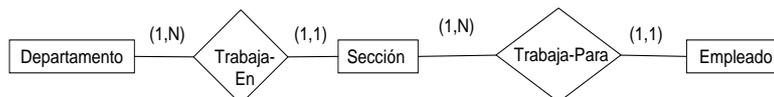


Figura 3.19: El modelo de la Figura 3.17 reestructurado.

Ahora ya es posible contestar a la pregunta que nos hacíamos anteriormente. En la Figura 3.20, podemos observar fácilmente que el empleado e_1 trabaja en la Sección s_1 y en el Departamento d_1 .

3.6.2. La trampa del sumidero

La trampa del sumidero ocurre cuando existe un tipo de relación entre dos tipos de entidad, pero no existe camino entre algunas entidades.

La trampa del sumidero puede aparecer cuando hay uno o más tipos de relación donde los tipos de entidad tienen una participación parcial. En la Figura 3.21 se presenta un diagrama ER con una trampa del sumidero potencial. El diagrama representa el hecho de que un

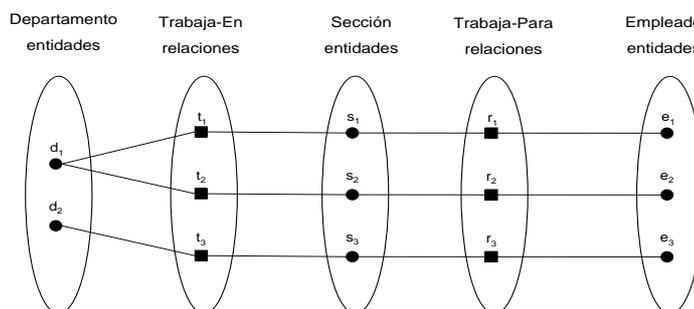


Figura 3.20: Ejemplo utilizando conjuntos de entidades e instancias de tipos de relación del modelo ER de la Figura 3.19.

Departamento tiene *uno o más* empleados que pueden dirigir proyectos, y no todos los proyectos son dirigidos por un empleado concreto. Suponemos que un proyecto (a diferencia de ejemplos anteriores) es realizado por un único departamento.

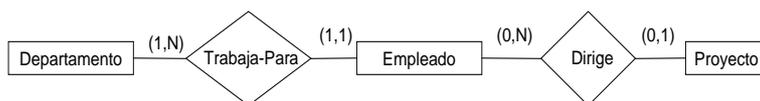


Figura 3.21: Ejemplo de la trampa del sumidero.

Para ilustrar el problema observar la Figura 3.22. Si realizamos la pregunta: “¿qué departamento realiza el proyecto p_2 ?”, no es posible obtener respuesta, dado que la entidad p_2 no está ligada a ningún empleado. Esta imposibilidad de contestar a la pregunta se conoce como una pérdida de información (dado que sabemos que el proyecto lo realiza un departamento), y el resultado es la trampa del sumidero. La participación de EMPLEADO y PROYECTO en el tipo de relación DIRIGE es parcial en ambos casos, lo que significa que algunos proyectos pueden no estar asociados a un departamento a través de un empleado. Por lo tanto para resolver este problema, necesitamos identificar la relación faltante, que en este caso es el tipo de relación REALIZA, entre los tipos de entidad DEPARTAMENTO y PROYECTO. El nuevo modelo ER se muestra en la Figura 3.23.

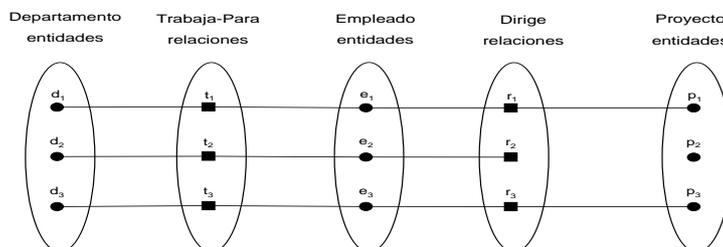


Figura 3.22: Ejemplo de conjuntos de entidades e instancias de relación para la Figura 3.21.

Si examinamos a las entidades e instancias de relación de la Figura 3.24, ahora es posible determinar que el proyecto p_2 es realizado por el departamento d_2

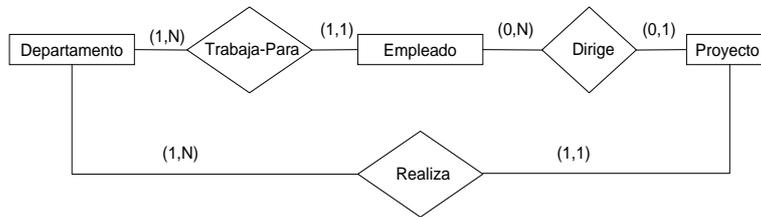


Figura 3.23: Ejemplo de la Figura 3.21 eliminando la trampa del sumidero.

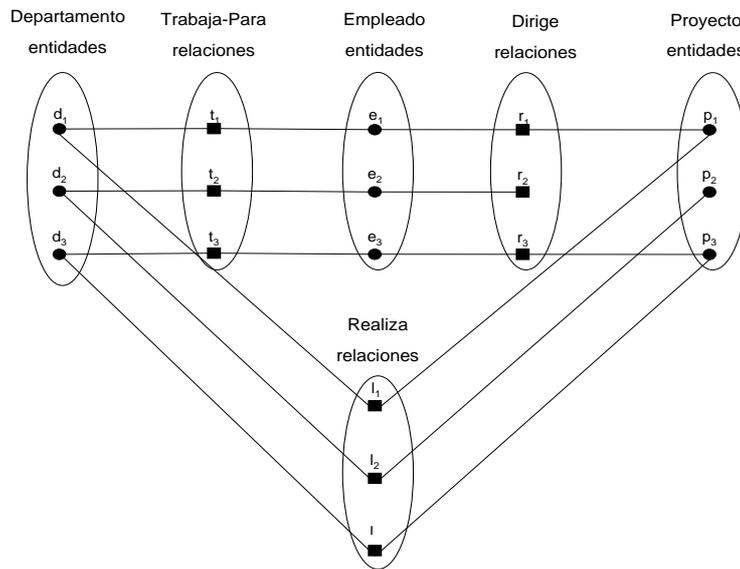


Figura 3.24: Ejemplo de la Figura 3.22 sin la trampa del sumidero.

3.7. Tipos de relación de grado superior a dos

En la Sección 3.4.2 definimos el grado de un tipo de relación como el número de tipos de entidad participantes y llamamos *binaria* a un tipo de relación de grado 2 y *ternaria* a un tipo de relación de grado 3. En esta sección, analizamos las diferencias entre las relaciones binarias y las de grado superior, cuándo elegir relaciones binarias o de grado superior y las restricciones en las relaciones de grado superior.

3.7.1. Elección entre las relaciones binarias y ternarias (o de grado superior)

La notación del diagrama ER para un tipo de relación ternaria se muestra en la Figura 3.25(a), que muestra el diagrama ER para el tipo de relación SUMINISTRO que se mostró a nivel de instancia en la Figura 3.7. En general, un tipo de relación R de grado n tendrá n líneas en un diagrama ER, que conectan R a cada tipo de entidad participante.

La Figura 3.25(b) muestra un diagrama ER para los tres tipos de relaciones binarias PUEDE-SUMINISTRAR, UTILIZA y SUMINISTRA. PUEDE-SUMINISTRAR incluye una instancia (s, p) siempre que el suministrador s pueda suministrar el componente p (a cualquier proyecto);

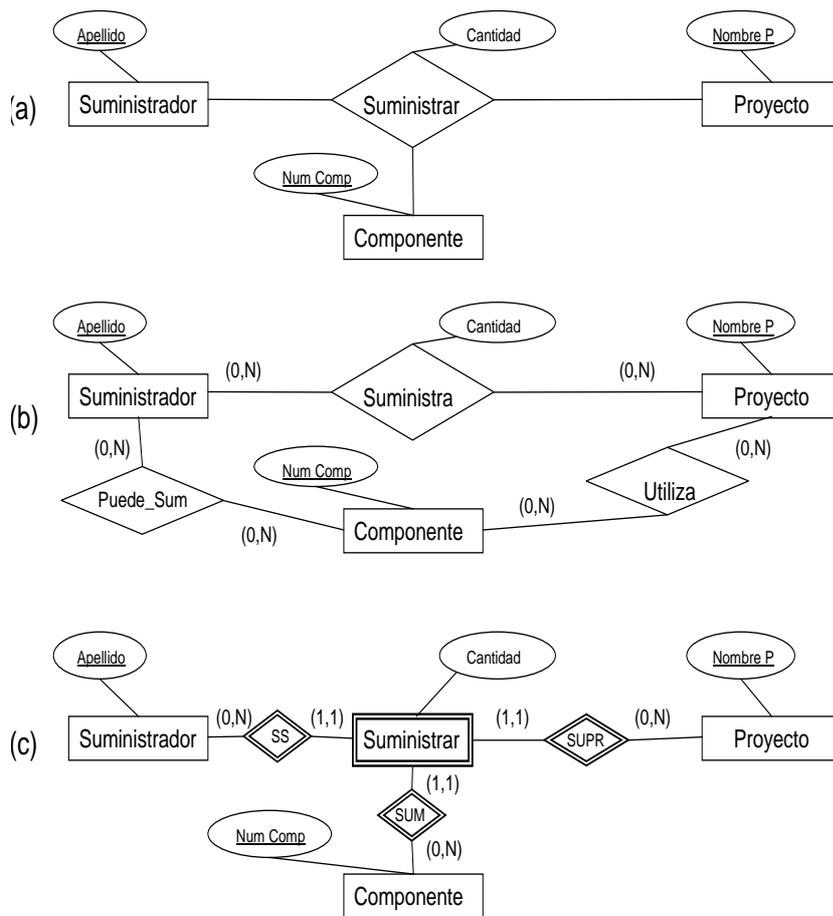


Figura 3.25: (a) Tipo de relación ternaria. (b) Tres tipos de relación binaria que no son equivalentes al tipo de relación ternaria. (c) La relación ternaria representada por un tipo de entidad débil.

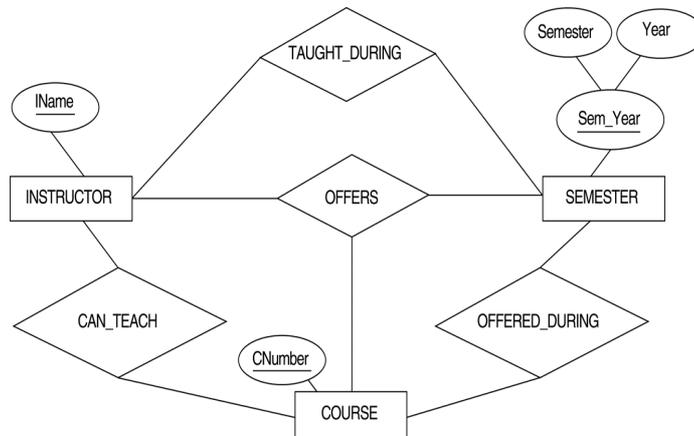


Figura 3.26: Otro ejemplo de relación ternaria frente a binarias

UTILIZA incluye una instancia (j, p) siempre que el proyecto j emplee componentes p y SUMINISTRA incluye una instancia (s, j) siempre que el suministrador s suministre algún componente al proyecto j . La existencia de las tres instancias de relación (s, p) , (j, p) y (s, j) en PUEDE-SUMINISTRAR, UTILIZA y SUMINISTRA, respectivamente, no implica necesariamente que una instancia (s, j, p) existan en la relación ternaria SUMINISTRAR ya que ¡el significado es diferente! A menudo es complicado decidir si una relación particular debería ser representada como un tipo de relación de grado n o si debería descomponerse en varios tipos de relación de grados inferiores. El diseñador debe basar su decisión en la semántica o en el significado de la situación particular que se representa. La solución típica es incluir la relación ternaria junto con una o más relaciones binarias, según se necesite.

Algunas de las herramientas de diseño de bases de datos se basan en variaciones del modelo ER que permiten únicamente relaciones binarias. En este caso, una relación ternaria como SUMINISTRAR debe representarse como un tipo de entidad débil, sin clave parcial y con tres relaciones identificadoras. Los tres tipos de entidad participantes PUEDE-SUMINISTRAR, UTILIZA y SUMINISTRA son todos ellos juntos los tipos de entidad propietarios (véase Figura 3.25(c)). Así pues, una entidad en el tipo de entidad débil SUMINISTRAR se identifica mediante la combinación de las claves primarias de sus tres entidades propietarias PUEDE-SUMINISTRAR, UTILIZA y SUMINISTRA.

En la Figura 3.26 se muestra otro ejemplo. El tipo de relación OFRECE representa información sobre los cursos (asignaturas) que imparten los profesores durante cuatrimestres (semestre en la figura), por tanto, incluye una instancia de relación (p, c, a) siempre que el profesor p imparta la asignatura a durante el cuatrimestre c . Los tres tipos de relación binarias mostrados en la Figura 3.26 tienen el siguiente significado: PUEDE-IMPARTIR relaciona una asignatura con el profesor que *puede impartir* esa asignatura; IMPARTIDO-DURANTE relaciona un cuatrimestre con los profesores que *impartieron alguna asignatura* durante ese cuatrimestre; y OFRECIDO-DURANTE relaciona un cuatrimestre con las asignaturas impartidas durante ese cuatrimestre por *cualquier profesor*. En general, dichas relaciones binarias y ternarias representan informaciones diferentes, pero deberían mantenerse ciertas restricciones entre las relaciones. Por ejemplo, una instancia de relación (p, c, a) no debería existir en OFRECE (la relación ternaria) a no ser que exista la instancia (p, a) en PUEDE-IMPARTIR, la instancia

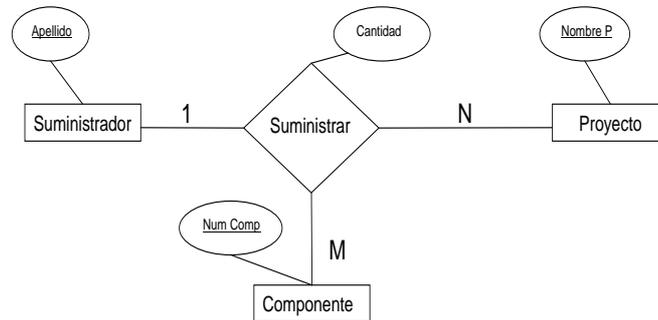


Figura 3.27: Relación ternaria con cardinalidad.

(p, c) en IMPARTIDO-DURANTE y la instancia (c, a) en OFRECIDO-DURANTE. Sin embargo, la relación opuesta no siempre es verdadera, puede ser que tengamos instancias (p, a) , (p, c) y (c, a) en los tres tipos de relación binarias y no exista la instancia (p, c, a) en OFRECE. Esto último puede suceder bajo ciertas *restricciones adicionales*, por ejemplo, si la relación PUEDE-IMPARTIR fuese 1:1. El diseñador del esquema debe analizar cada situación particular para decidir qué tipos de relación binaria o ternaria se necesitan.

3.7.2. Restricciones sobre relaciones ternarias (o de grado superior)

Un (mín, máx) sobre una participación especifica que cada entidad está relacionada con por lo menos *mín* y como mucho *max* instancias de la relación en el conjunto de la relación. Esta notación no es muy útil, porque lo normal es que una entidad esté en muchas instancias de la relación, en otro caso no tendría mucho sentido una relación ternaria.

Usando la notación alternativa que presentamos en la Sección 3.4.7. Usando el ejemplo de la Figura 3.25, supongamos que la relación que existe, es que para una combinación particular de proyecto-componente, sólo se puede usar un suministrador (un único suministrador suministra un componente particular a un proyecto determinado). En este caso, colocamos un 1 en la participación de SUMINISTRADOR, y M, N en las participaciones de PROYECTO y COMPONENTE, resultando la Figura 3.27.

Esto especifica la restricción de que una combinación particular (j, p) puede aparecer *como mucho una vez* en el conjunto de las relaciones. Así pues, cualquier instancia de la relación (s, j, p) está *identificada de forma única* en el conjunto de la relación por su combinación (j, p) , lo cual hace de (j, p) una clave para el conjunto de la relación. En general, las participaciones que tienen un 1 especificado sobre ellas no se requiere que sean parte de la clave del conjunto de la relación.