

# Fundamentos de Bases de Datos



UNIVERSIDAD  
DE GRANADA

*Escuela Técnica Superior de Ingenierías  
Informática y de Telecomunicación*

Los Del DGIIM, [losdelgiim.github.io](https://losdelgiim.github.io)

Doble Grado en Ingeniería Informática y Matemáticas  
Universidad de Granada



Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-SinDerivadas 4.0 Internacional (CC BY-NC-ND 4.0).

Eres libre de compartir y redistribuir el contenido de esta obra en cualquier medio o formato, siempre y cuando des el crédito adecuado a los autores originales y no persigas fines comerciales.

# Fundamentos de Bases de Datos

Los Del DGIIM, [losdeldgiim.github.io](https://losdeldgiim.github.io)

Arturo Olivares Martos

Granada, 2024-2025



# Índice general

<b>1. Seminarios</b>	<b>5</b>
1.1. Modelo Entidad-Relación . . . . .	5
1.2. Álgebra Relacional . . . . .	34



# 1. Seminarios

## 1.1. Modelo Entidad-Relación

**Ejercicio 1.1.1.** Queremos crear la BD para una biblioteca:

- Los libros se caracterizan por su ISBN, título y año de escritura.
- Los autores tienen código, nombre y nacionalidad.
- No existe más que un ejemplar de cada libro.
- Cada libro puede estar escrito por más de un autor.
- Un autor puede escribir más de un libro.
- Cada libro puede tratar más de un tema.
- Hay muchos libros de cada tema.
- Los usuarios de la biblioteca están caracterizados por su DNI, su nombre y su dirección.
- Queremos poder representar información relativa a los préstamos. Para ello registramos cuándo un usuario toma prestado un libro y eliminamos dicho registro cuando el usuario lo devuelve. Registramos también la fecha del préstamo.
- Cada usuario no puede tener prestado más de un libro simultáneamente.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.1. Notemos que ambas participaciones obligatorias no están explícitamente en el enunciado, aunque se pueden inferir del contexto en el que trabajamos. Esto se tendría que especificar con el cliente. También es importante notar que, como la entrada del préstamo se borra, la fecha no es discriminante.

El paso a tablas está en la Figura 1.2. Notemos que la tabla *Préstamo* podría fusionarse con *Libro*, pero no lo hemos hecho puesto los préstamos se añaden y se borrarán, mientras que los libros han de permanecer.

**Ejercicio 1.1.2.** Considere el Ejercicio 1.1.1, con las siguientes modificaciones:

- Existen varios ejemplares de cada libro.
- Se registra información sobre el histórico de préstamos que sufre un libro.

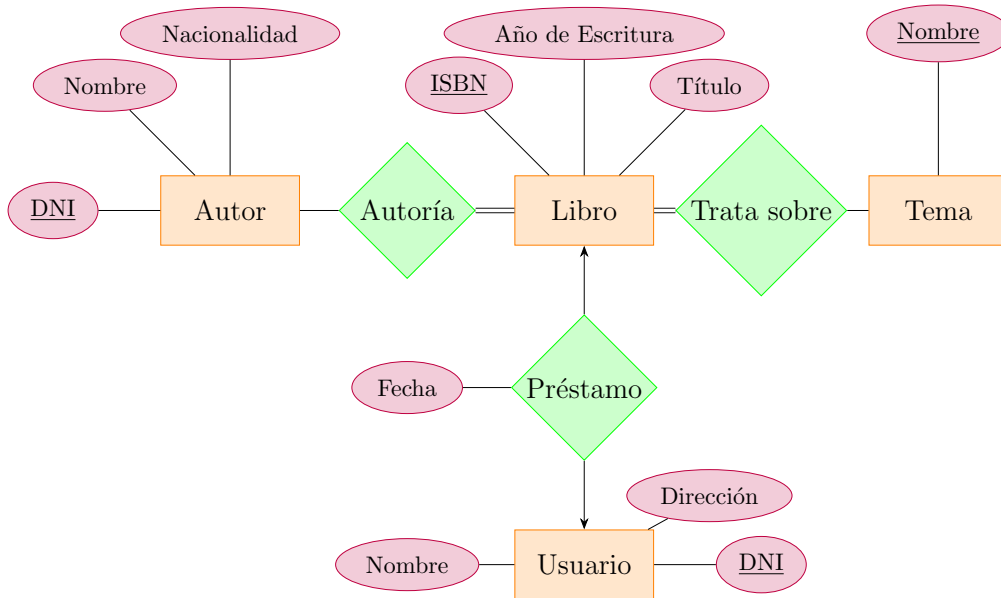


Figura 1.1: Diagrama Entidad-Relación del Ejercicio 1.1.1.

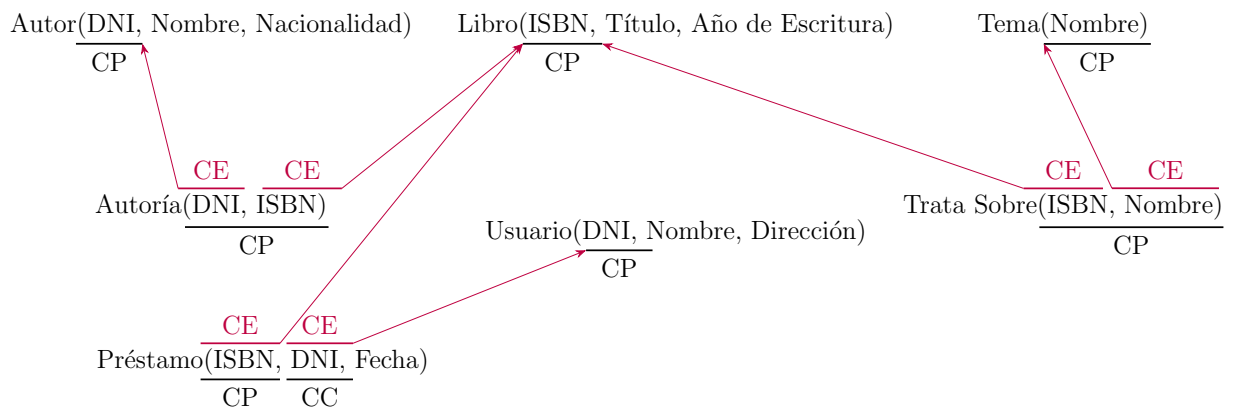


Figura 1.2: Tablas de Modelo E-R del Ejercicio 1.1.1.



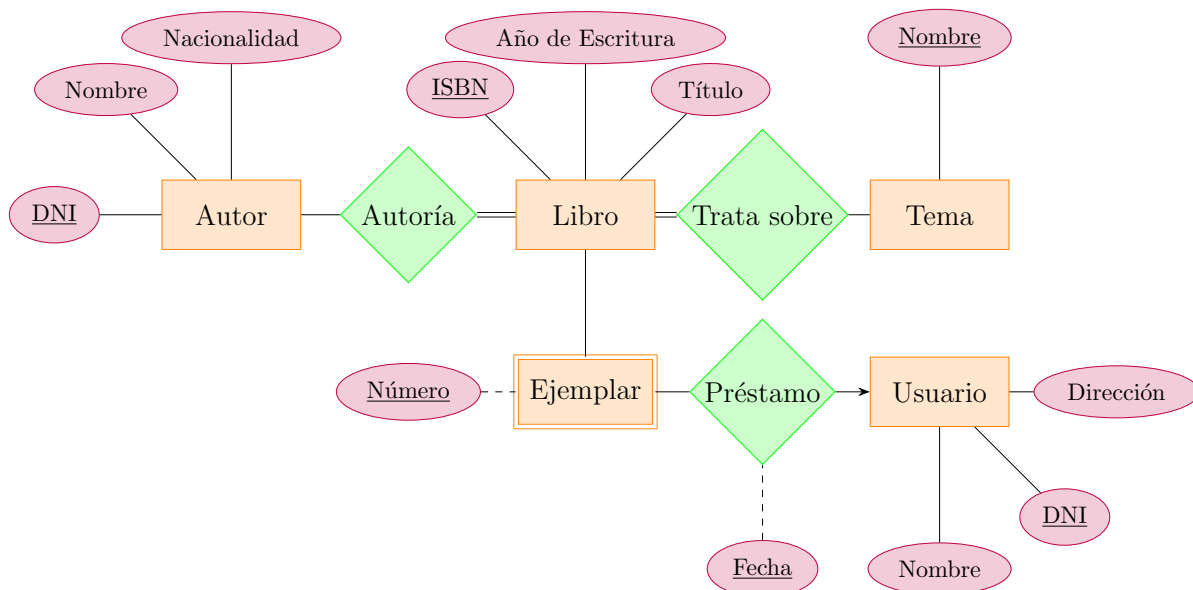


Figura 1.3: Diagrama Entidad-Relación del Ejercicio 1.1.2.

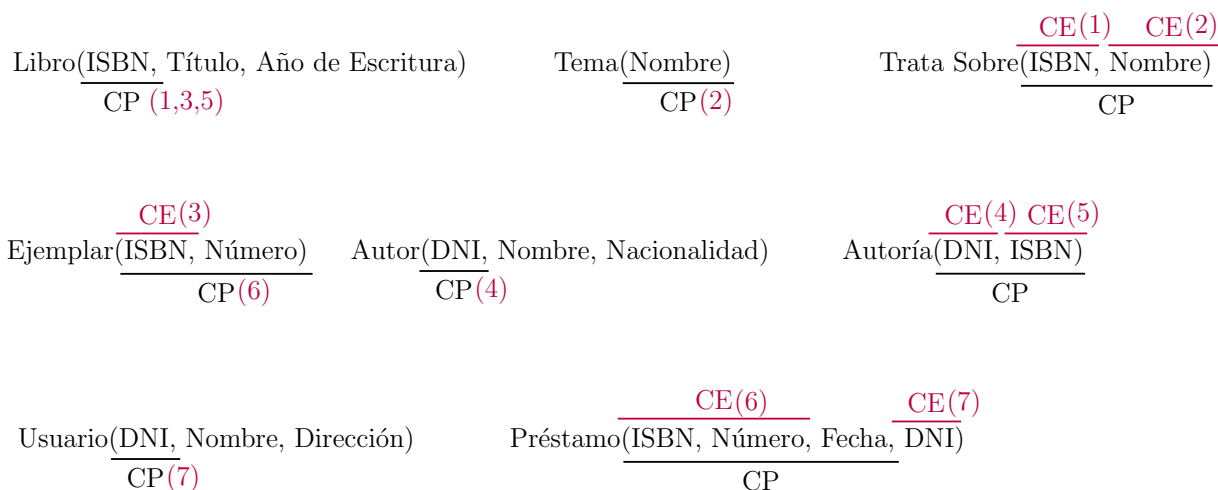


Figura 1.4: Tablas de Modelo E-R del Ejercicio 1.1.2.

- Un usuario puede tener prestados varios libros al mismo tiempo.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.3. El paso a tablas está en la Figura 1.4.

**Ejercicio 1.1.3.** Se quiere hacer una BD para una empresa de alquiler de DVDs, considerando las siguientes restricciones semánticas:

- Las películas están caracterizadas por su título, año de estreno, actores principales y tema.
- De los clientes se almacena su DNI, nombre, dirección y teléfono.
- Puede haber películas distintas con el mismo nombre (versiones), pero estas deben ser de distinto año.

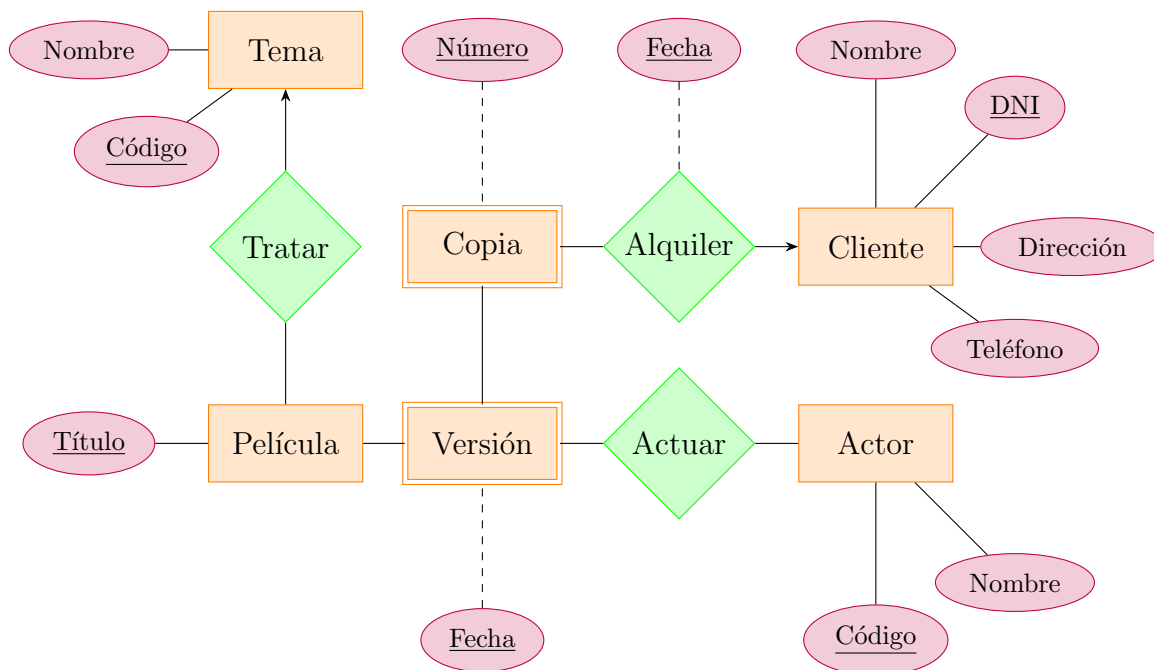


Figura 1.5: Diagrama Entidad-Relación del Ejercicio 1.1.3 y 1.1.4.

- Hay distintas copias de cada película que se pueden alquilar.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.5.

El paso a tablas está en la Figura 1.6, donde la fusión se ha realizado por debajo de la línea divisoria discontinua.

**Ejercicio 1.1.4.** Considere el ejercicio 1.1.3 y la siguiente restricción adicional:

- Las películas con el mismo título tienen el mismo tema.

Notemos que la restricción ya estaba establecida en el ejercicio 1.1.3, ya que hemos creado la entidad débil de versión. El ejercicio anterior se podría haber resuelto de otras maneras, aunque dificultarían escalarlo. Las tablas resultantes se encuentran también por tanto en la Figura 1.6.

**Ejercicio 1.1.5.** Se quiere gestionar información relativa a la publicación de artículos científicos en revistas:

- Una revista se identifica por un ISSN y tiene un nombre y editorial.
- Durante un año la revista publica uno o varios números que recogen los artículos aceptados. De cada número de la revista se recoge la fecha de publicación.
- Cada número contiene uno o varios artículos.
- Cada artículo tiene un título y una lista ordenada de autores.
- También se almacena la página de inicio y de fin en el número de la revista en el que se ha publicado.
- Cada autor se identifica por un código y se caracteriza por su nombre y nacionalidad.

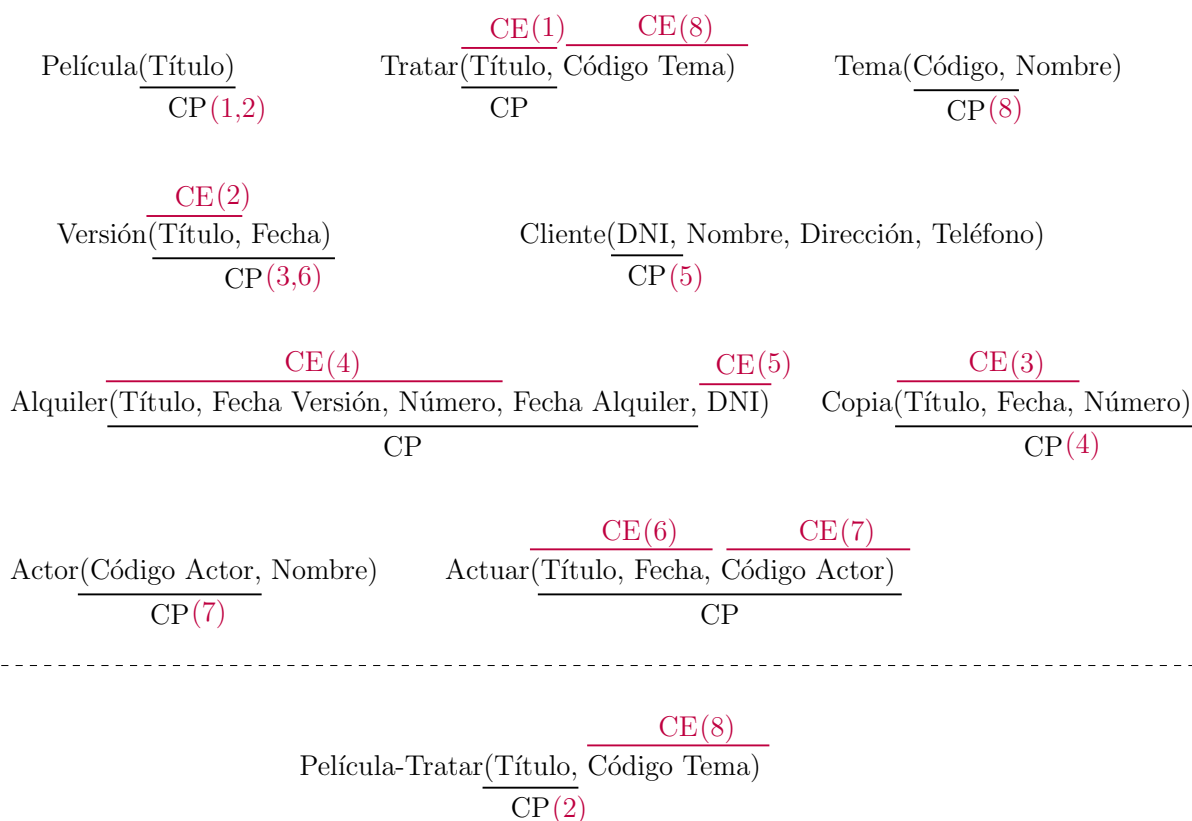


Figura 1.6: Tablas de Modelo E-R del Ejercicio 1.1.3.

- Un artículo puede estar escrito por varios autores y un autor puede escribir varios artículos.
- Un artículo puede hacer referencia a otros artículos y puede ser citado en otros artículos.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.7. Notemos que, para poder almacenar el orden, hemos añadido el atributo *orden* en la relación *Escrito por*. Podríamos tener el problema de que, para un mismo artículo, dos autores tengan el mismo orden. Esto será algo que se deberá controlar en la programación.

El paso a tablas está en la Figura 1.8.

**Ejercicio 1.1.6.** Se quiere hacer una BD para registrar la información recogida en el modelo de factura de la Figura 1.9.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.10.

El paso a tablas está en la Figura 1.11, donde la fusión se ha realizado por debajo de la línea divisoria discontinua.

**Ejercicio 1.1.7.** En una empresa mecánica se quiere poder calcular el precio de las piezas instaladas en un coche, sabiendo que algunas de las piezas pueden tener varios componentes. Para ello debemos representar cada tipo de pieza, del que se registra un código y su denominación. Se supone que:

- Hay dos tipos de pieza simple o compuesta.

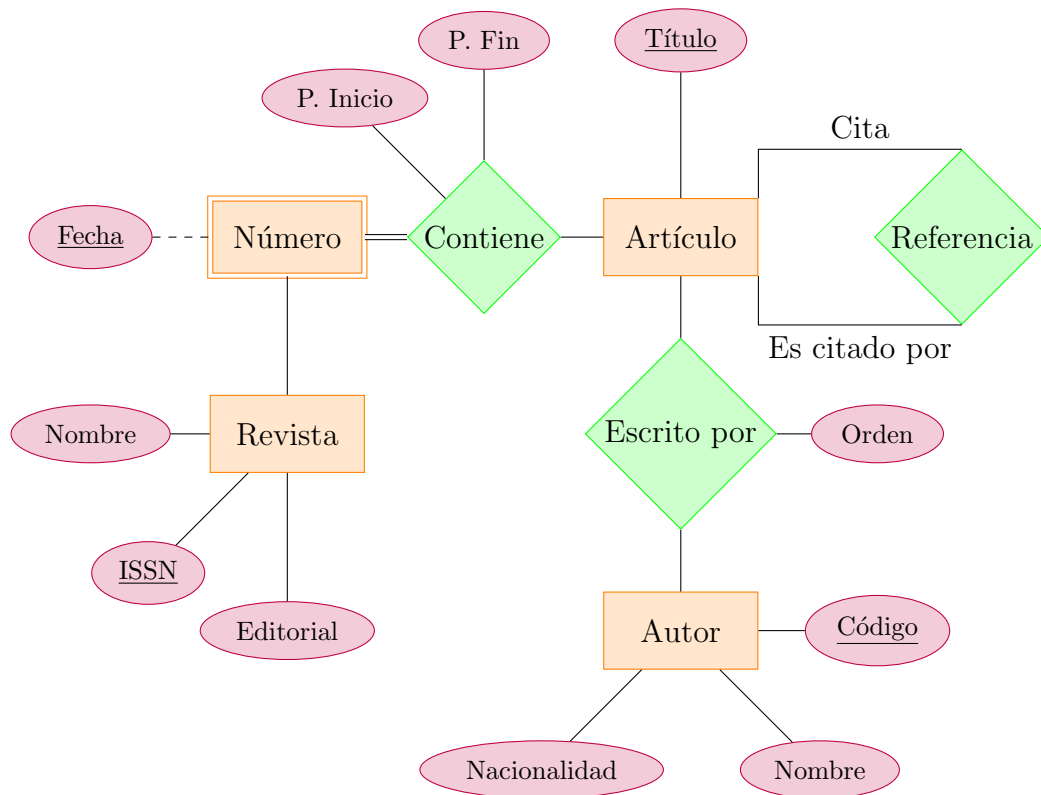


Figura 1.7: Diagrama Entidad-Relación del Ejercicio 1.1.5.

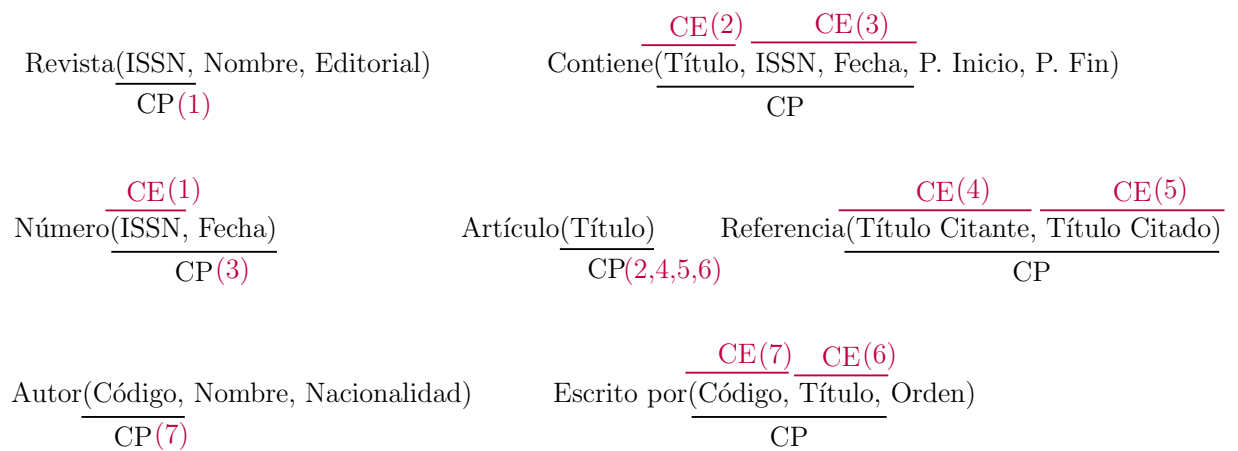


Figura 1.8: Tablas de Modelo E-R del Ejercicio 1.1.5.

Empresa Vendedora S.A.		Factura Nº _____		
A1234567B				
C/Mi sede, nº 1				
Nombre Cliente Comprador				
12345678A				
C/El domicilio, nº 2				
Nº línea	Artículo	Cantidad	Precio	Total
SUBTOTAL				
IMPUESTOS (21%)				
IMPORTE TOTAL				

Figura 1.9: Modelo de Factura del Ejercicio 1.1.6.

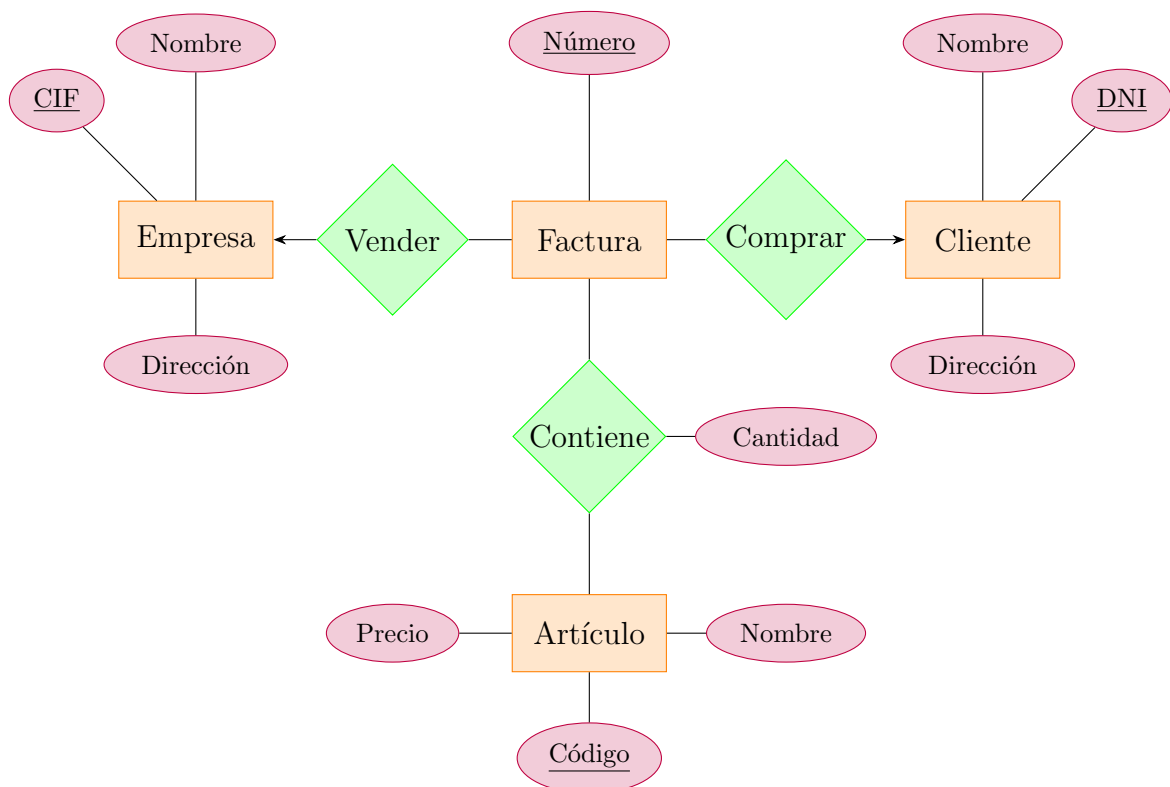


Figura 1.10: Diagrama Entidad-Relación del Ejercicio 1.1.6.

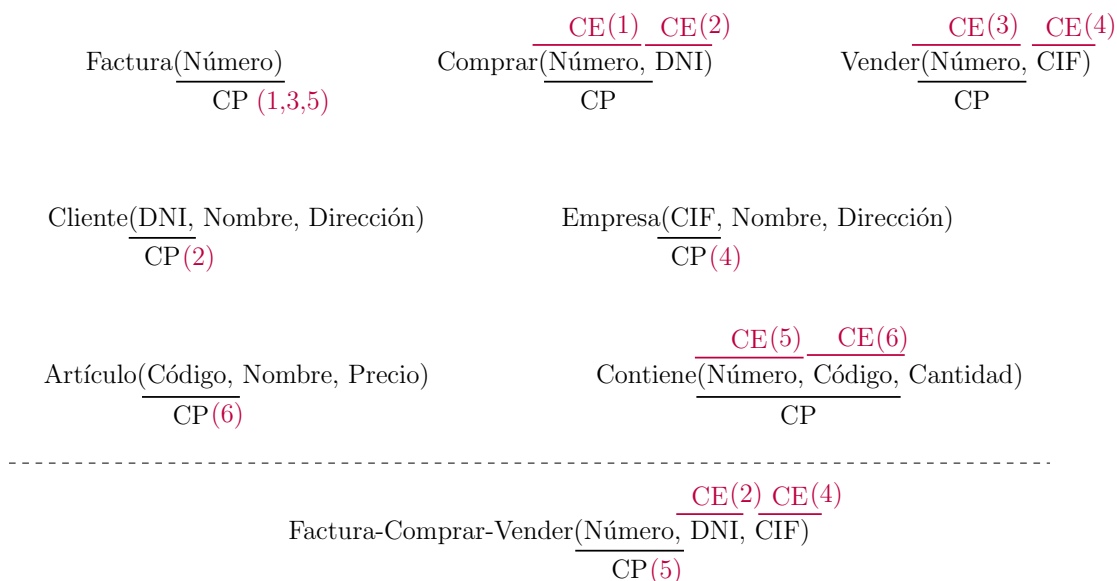


Figura 1.11: Tablas de Modelo E-R del Ejercicio 1.1.6.

- El precio de un tipo de pieza simple consiste en el valor dicha pieza.
- Si el tipo de pieza es compuesto su precio se corresponde con el precio de montaje sin incluir el precio de los tipos de pieza que la componen.
- Para los tipos de pieza compuestos se registran el número de unidades de cada tipo de pieza que la compone.
- Un tipo de pieza es componente de un único tipo de pieza compuesta.

Resuelva el problema considerando:

1. Que las piezas compuestas solo pueden estar compuestas de piezas simples.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.12.

El paso a tablas está en la Figura 1.13, donde la fusión se ha realizado por debajo de la línea divisoria discontinua. Notemos que, aunque se hayan fusionado las tablas *Simple-Componente*, esto posiblemente no sería lo más eficiente debido a que, todas las piezas que no fuesen componente de ninguna otra, tendrían un valor nulo en los atributos *Unidades* y *Código Compuesta*.

2. Que las piezas compuestas pueden estar compuestas tanto de piezas simples como compuestas.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.14.

Las tablas resultantes se encuentran en la Figura 1.15, donde la fusión se ha realizado por debajo de la línea divisoria discontinua. Notemos que, aunque se hayan fusionado las tablas *Pieza-Componente*, esto posiblemente no sería lo más eficiente debido a que, todas las piezas que no fuesen componente de ninguna otra, tendrían un valor nulo en los atributos *Unidades* y *Código*.

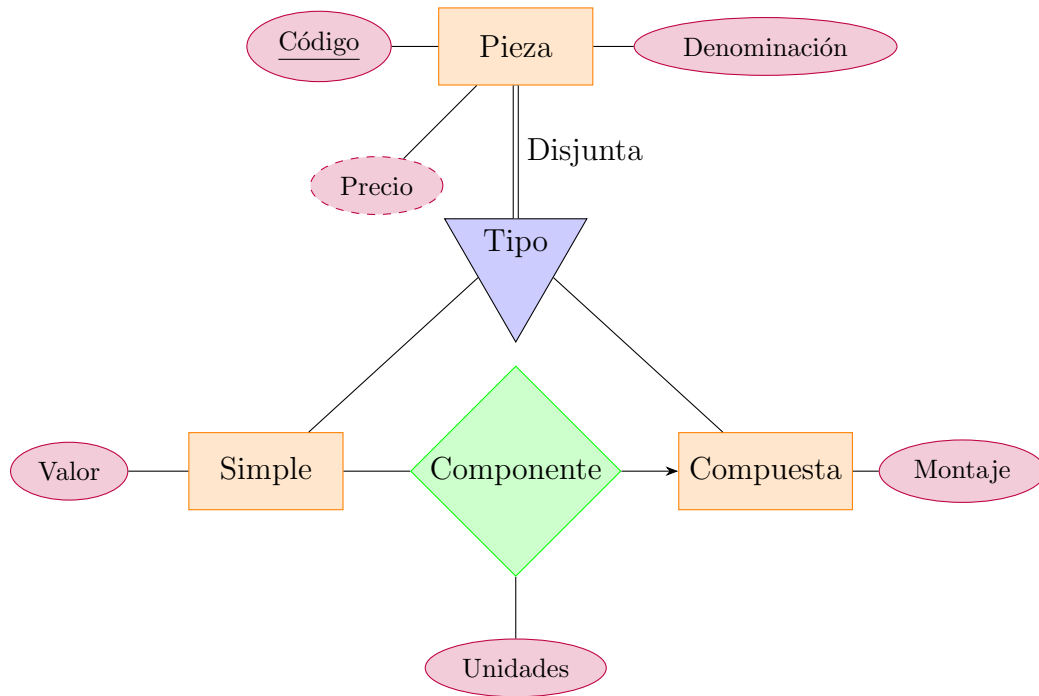


Figura 1.12: Diagrama Entidad-Relación del Ejercicio 1.1.7 del apartado 1.

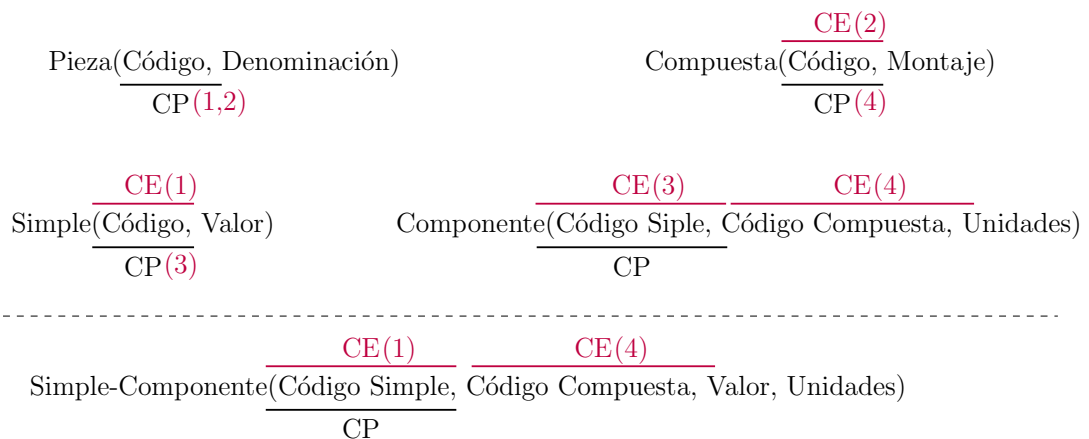


Figura 1.13: Tablas de Modelo E-R del Ejercicio 1.1.7 del apartado 1.

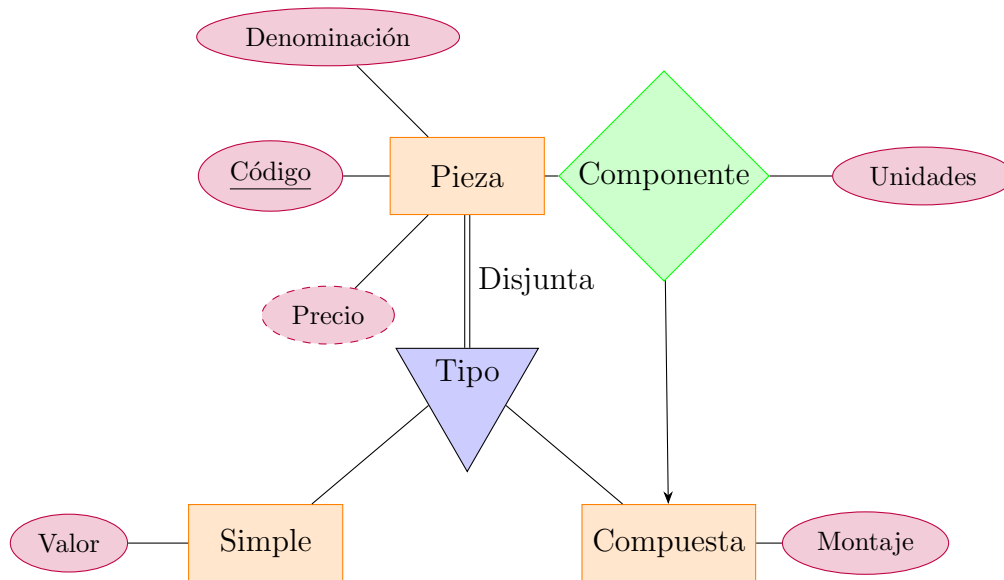


Figura 1.14: Diagrama Entidad-Relación del Ejercicio 1.1.7 del apartado 2.

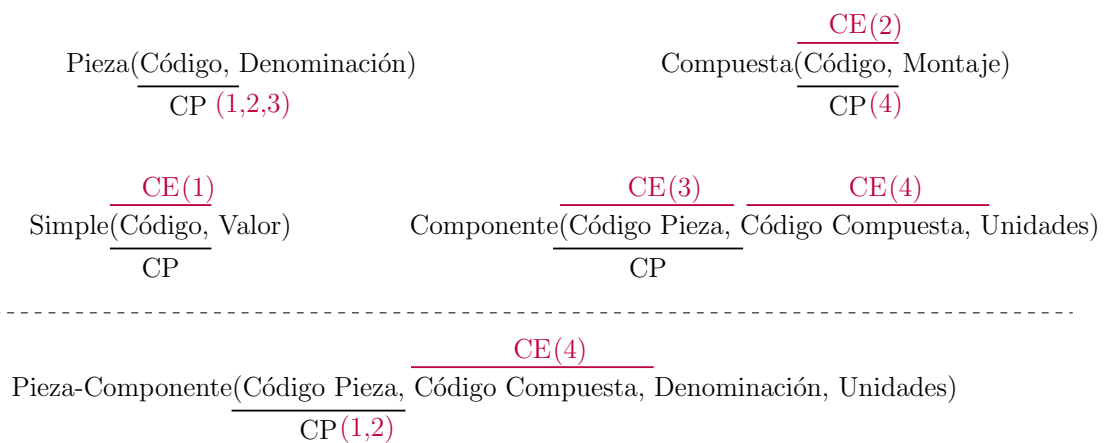


Figura 1.15: Tablas de Modelo E-R del Ejercicio 1.1.7 del apartado 2.



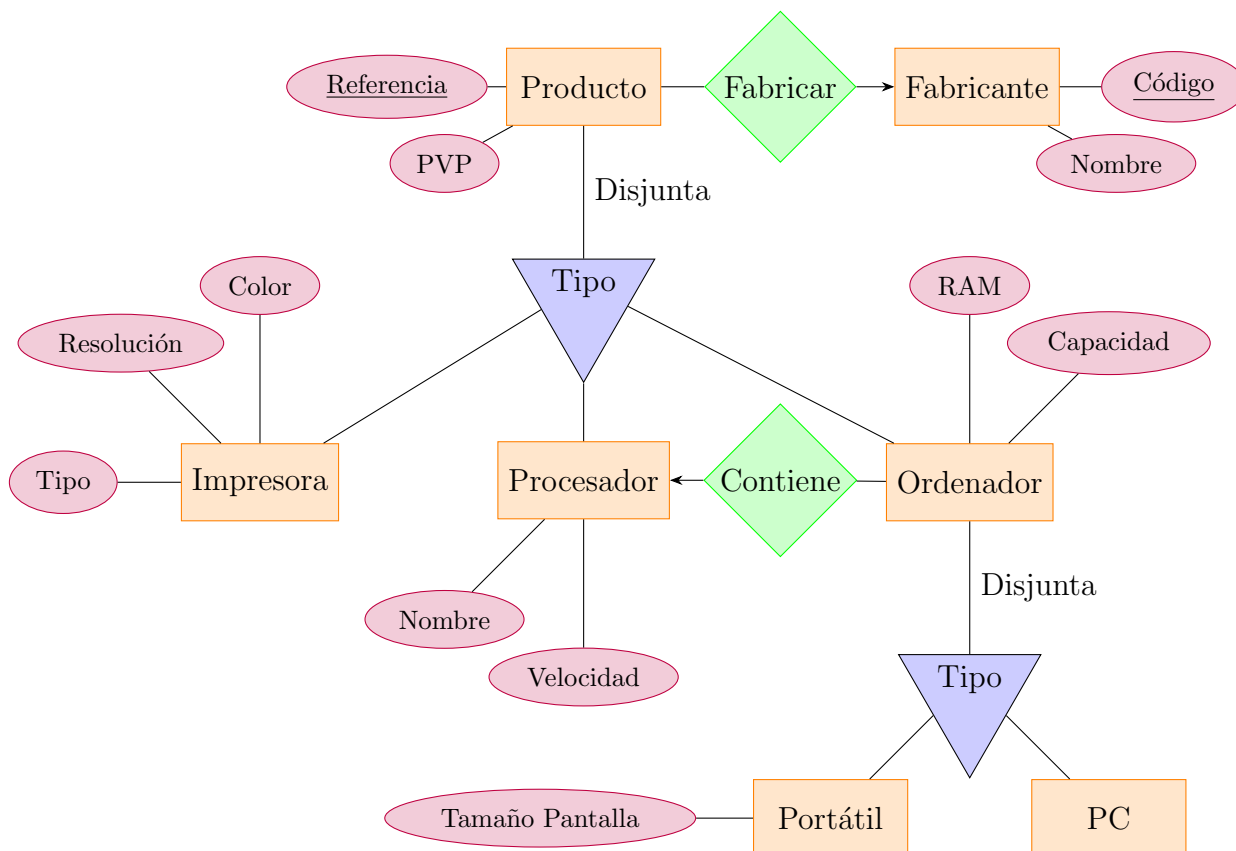


Figura 1.16: Diagrama Entidad-Relación del Ejercicio 1.1.8.

**Ejercicio 1.1.8.** En una base de datos de una tienda de productos informáticos, los tipos de productos se registran con un número de referencia, un fabricante y tienen un precio de venta al público. De los artículos estrella de la tienda (impresoras, PCs y portátiles) se registran las siguientes características adicionales:

- Impresoras: color (s/n), resolución (ppp), tipo (láser o inyección de tinta).
- PC: procesador, velocidad, RAM, capacidad del disco.
- Portátiles: procesador, velocidad, RAM, capacidad del disco, tamaño de pantalla.

Considere que cada procesador tiene una velocidad determinada.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.16. Notemos que, al incluir un procesador como un tipo de producto, que se pueden vender estos también por separado.

El paso a tablas está en la Figura 1.17, donde la fusión de tablas se ha realizado por debajo de la línea divisoria.

**Ejercicio 1.1.9.** Expresar mediante un diagrama E/R el registro de llamadas entre dos teléfonos, conteniendo fecha y hora de inicio y de finalización. Supongamos que un teléfono se identifica mediante un número y que podemos contar con dos tipos de teléfono: fijo o móvil. No se contemplan llamadas en las que participen más de dos teléfonos.

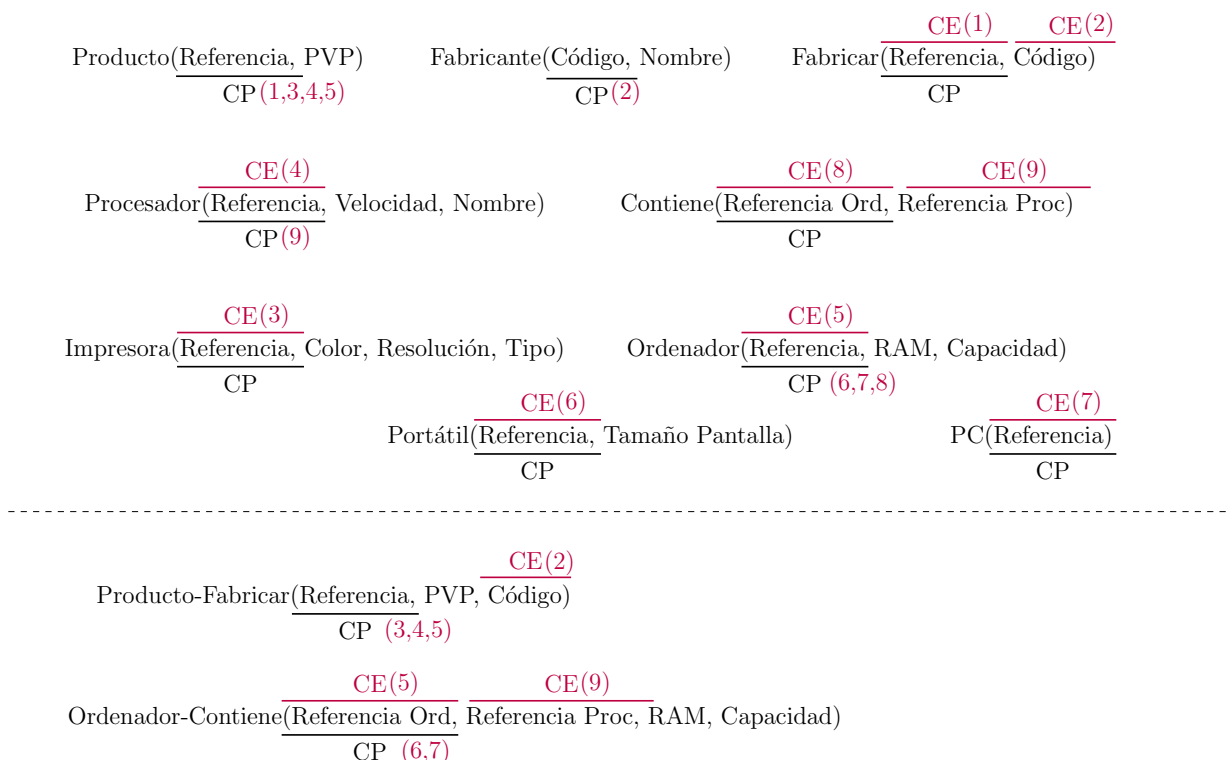


Figura 1.17: Tablas de Modelo E-R del Ejercicio 1.1.8.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.18.

El paso a tablas está en la Figura 1.19. No se ha realizado fusión puesto que esta no es posible. Además, como hay tres claves candidatas posibles para la misma tabla, se han establecido a distinta altura.

**Ejercicio 1.1.10.** Una receta de cocina se describe mediante una serie de ingredientes y de pasos de ejecución. En cada paso de ejecución se describe una acción y se utilizan una serie de ingredientes. Las recetas se caracterizan por:

- Código; Nombre; Tipo (primero, segundo y postre) y dificultad (alta, media, baja).
- A su vez, los ingredientes se caracterizan por: Código; Nombre; Tipo (grano, polvo, troceado y otro) y precio.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.20. Notemos que los atributos *Tipo* de *Receta* e *Ingrediente* se podrían haber modelado como jerarquías.

El paso a tablas está en la Figura 1.21. De nuevo, no se realiza fusión de tablas por no ser posible.

**Ejercicio 1.1.11.** Se trata de modelar la programación que ofrecen los canales de TV. La información que se desea almacenar es la siguiente:

- Presentador (DNI, Nombre, Especialidad).
- Canales (Nombre, Empresa, Tipo).

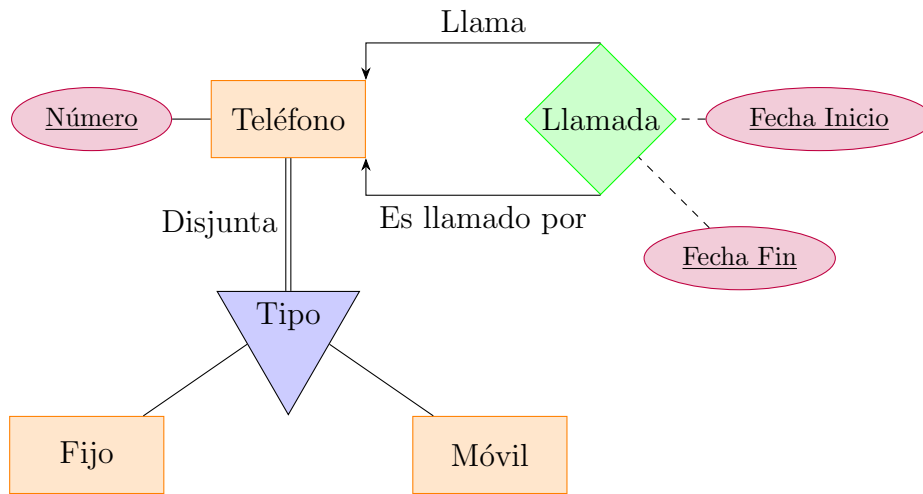


Figura 1.18: Diagrama Entidad-Relación del Ejercicio 1.1.9.

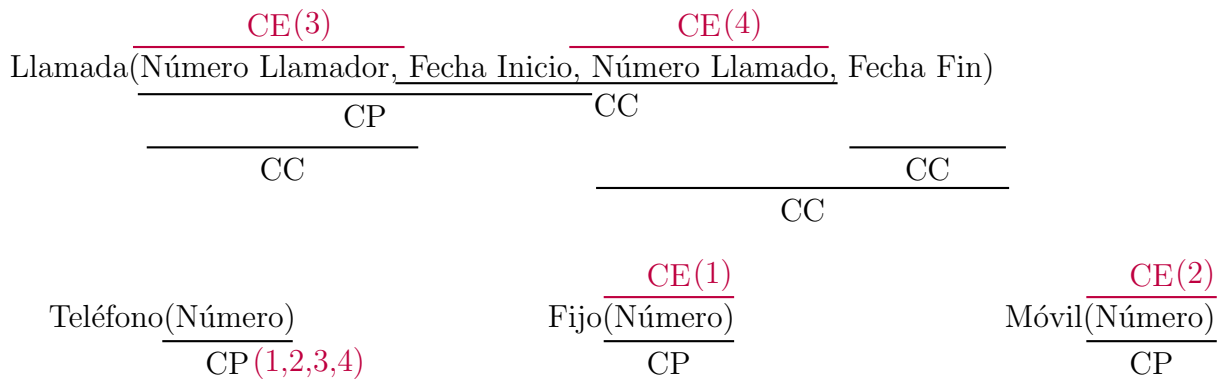


Figura 1.19: Tablas de Modelo E-R del Ejercicio 1.1.9.

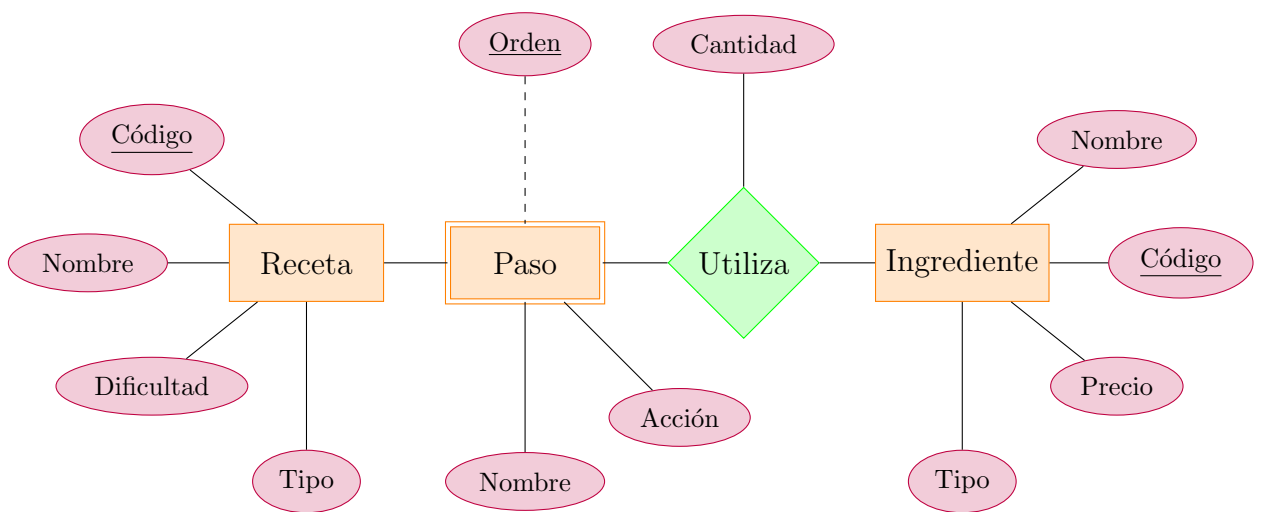


Figura 1.20: Diagrama Entidad-Relación del Ejercicio 1.1.10.

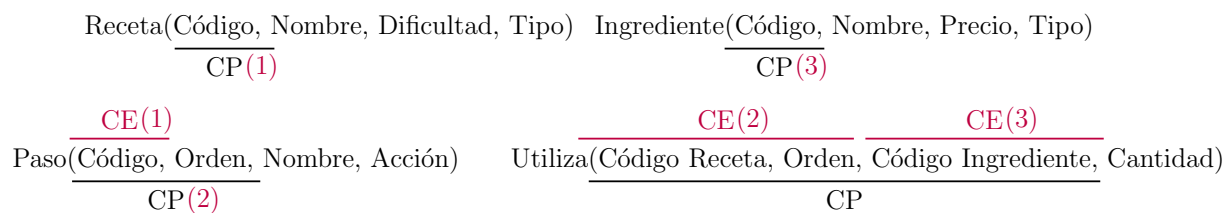


Figura 1.21: Tablas de Modelo E-R del Ejercicio 1.1.10.

- Programas (Código, Nombre, Duración, Tipo).

Las restricciones de integridad que deben mantenerse son las siguientes:

- Existen tres tipos de programas:
  - Películas, de las que hay que conocer Título, Tema y Año.
  - Concursos.
  - Informativos.
- En cada momento, solo hay un programa en emisión en cada canal.
- Los concursos solo pueden ser presentados por un presentador, pero un presentador puede serlo de varios concursos.
- Los informativos pueden ser presentados por varios presentadores y un presentador puede serlo de varios también.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.22. Respecto a la condición de que solo haya un programa en emisión en cada canal, se ha añadido el atributo *Fecha* en la relación *Emite*. Esto provoca que, en un canal, dos programas no puedan comenzar a emitirse en la misma fecha, aunque no evita que, debido a la duración, se solapen. Esto último habrá que controlarlo en la programación.

El paso a tablas está en la Figura 1.23. La fusión de tablas se ha realizado por debajo de la línea divisoria discontinua. Notemos que  $CC(*)$  denota que *Título* no se puede repetir, pero no es totalmente una clave candidata.

**Ejercicio 1.1.12.** Diseña una BD que gestione los datos que se manejan en las comisarías de policía de una ciudad. En dicha BD debe registrarse la siguiente información:

- DELINCUENTES, que van identificados por su DNI y de los que debe conocerse Nombre y Apellidos, Fecha de Nacimiento, Domicilio y Nacionalidad.
- DELITOS, identificados por un código y de los que hay que registrar su Nombre, Tipo, y Penas Máxima y Mínima.
- AGENTES, identificados por su Número de Placa y de los que debe conocerse su DNI, Nombre y Apellidos, Domicilio y Fecha de Ingreso en el cuerpo.
- COMISARÍAS, identificadas por un Nombre y de las que debe registrarse su Localización.

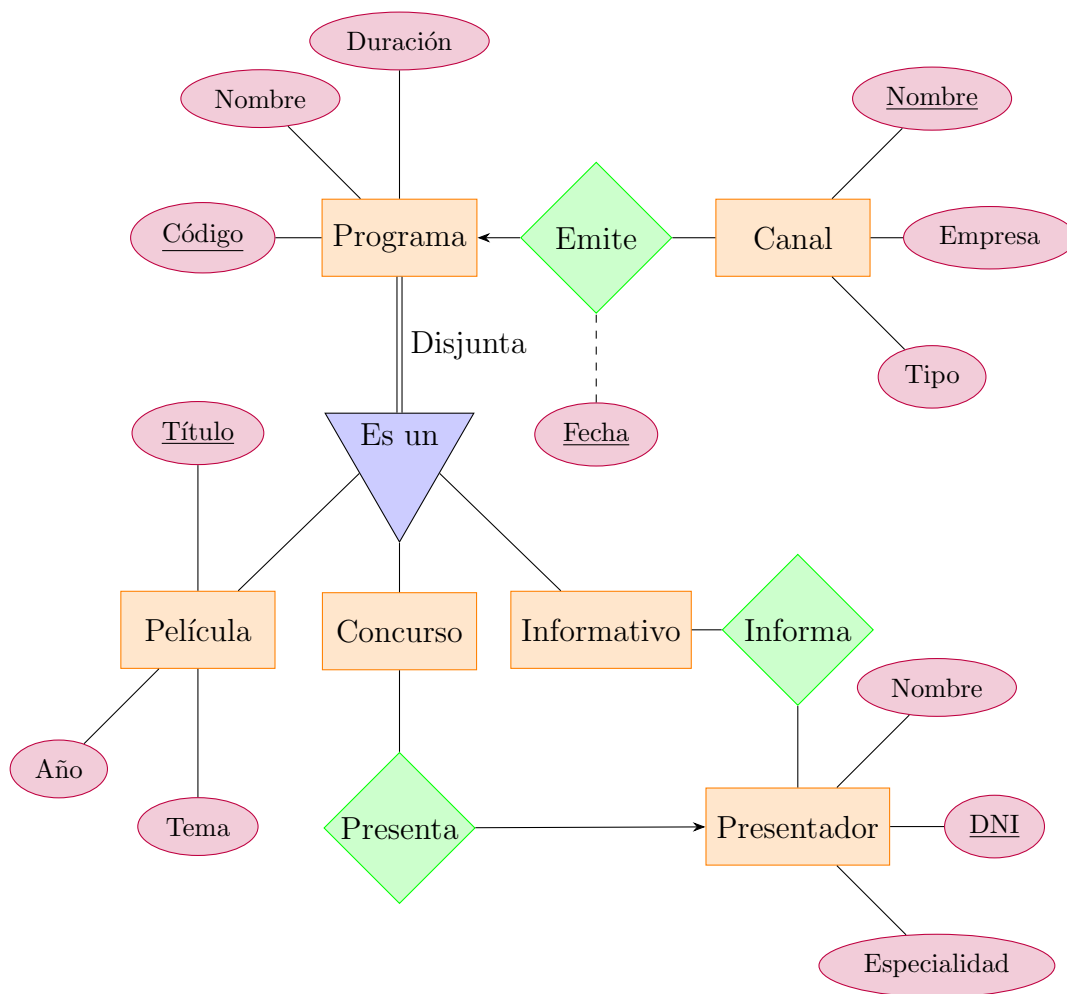


Figura 1.22: Diagrama Entidad-Relación del Ejercicio 1.1.11.

Programa( <u>Código</u> , Nombre, Duración) CP (1,2,3,4)		
Película( <u>Código</u> , <u>Título</u> , Tema, Año) CP CC(*)	Concurso( <u>Código</u> ) CP (6)	Informativo( <u>Código</u> ) CP (8)
Canal( <u>Nombre</u> , Empresa, Tipo) CP (5)	Emite( <u>Código Programa</u> , <u>Nombre Canal</u> , Fecha) CP	
Presentador( <u>DNI</u> , Nombre, Especialidad) CP (7,9)	Presenta( <u>Código</u> , <u>DNI</u> ) CP	Informa( <u>Código</u> , <u>DNI</u> ) CP
Concurso-Presenta( <u>Código</u> , <u>DNI</u> ) CP		

Figura 1.23: Tablas de Modelo E-R del Ejercicio 1.1.11.

Las restricciones semánticas que deben cumplirse son las siguientes:

- Existen dos tipos de AGENTES: OFICIALES y POLICIAS.
- De los oficiales debe conocerse la Especialización y los años de Experiencia.
- Un oficial es responsable de un grupo de policías y un policía solo depende de un oficial.
- Cada comisaría la dirige un oficial.
- Cuando un delincuente es detenido, debe registrarse el lugar, la fecha y la hora de la detención y qué agentes han participado en la misma.
- De los delitos cometidos, debe conocerse el lugar, la fecha y la hora, así como los delincuentes que han participado.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.24. Algunas consideraciones son:

- Se ha establecido la superclase *Persona* para *Agente* y *Delincuente*, ya que comparten varios atributos. Además, en el hipotético caso de que un agente sea detenido, no habría redundancia.
- La relación *Realiza* tiene cardinalidad  $M : N$ , ya que una comisión de un delito puede ser realizada por varios delincuentes, y un delincuente puede realizar varias comisiones de delito en su vida.
- La relación *Es* tiene cardinalidad  $M : N$ , ya que una comisión de delito puede ser de varios tipos (por ejemplo, en un atraco, puede haber robo, secuestro, allanamiento de morada, etc.), y un tipo de delito puede ser cometido en varias ocasiones.
- En un primer momento, podríamos haber establecido *Detención* como una relación entre *Agente* y *Delincuente*, pero los atributos asociados a la detención (lugar, fecha y hora) se repetirían en gran cantidad de ocasiones (si  $n$  agentes participan en la misma detención, se repetirían  $n$  veces). Por ello, se ha decidido crear una entidad propia para la detención.
- De igual forma, la comisión de un delito se ha considerado como una entidad propia, ya que, si no, si  $n$  delincuentes realizan el mismo delito, se repetirían  $n$  veces los atributos asociados a la comisión del delito.

El paso a tablas está en la Figura 1.25. La fusión de tablas se ha realizado por debajo de la línea divisoria discontinua.

**Ejercicio 1.1.13.** Queremos gestionar una base de datos que contenga información sobre objetos astronómicos:

- Un objeto astronómico se identifica mediante un código y, además, se registra la fecha y observatorio donde se hizo su descubrimiento.
- Los objetos astronómicos los vamos a clasificar en:

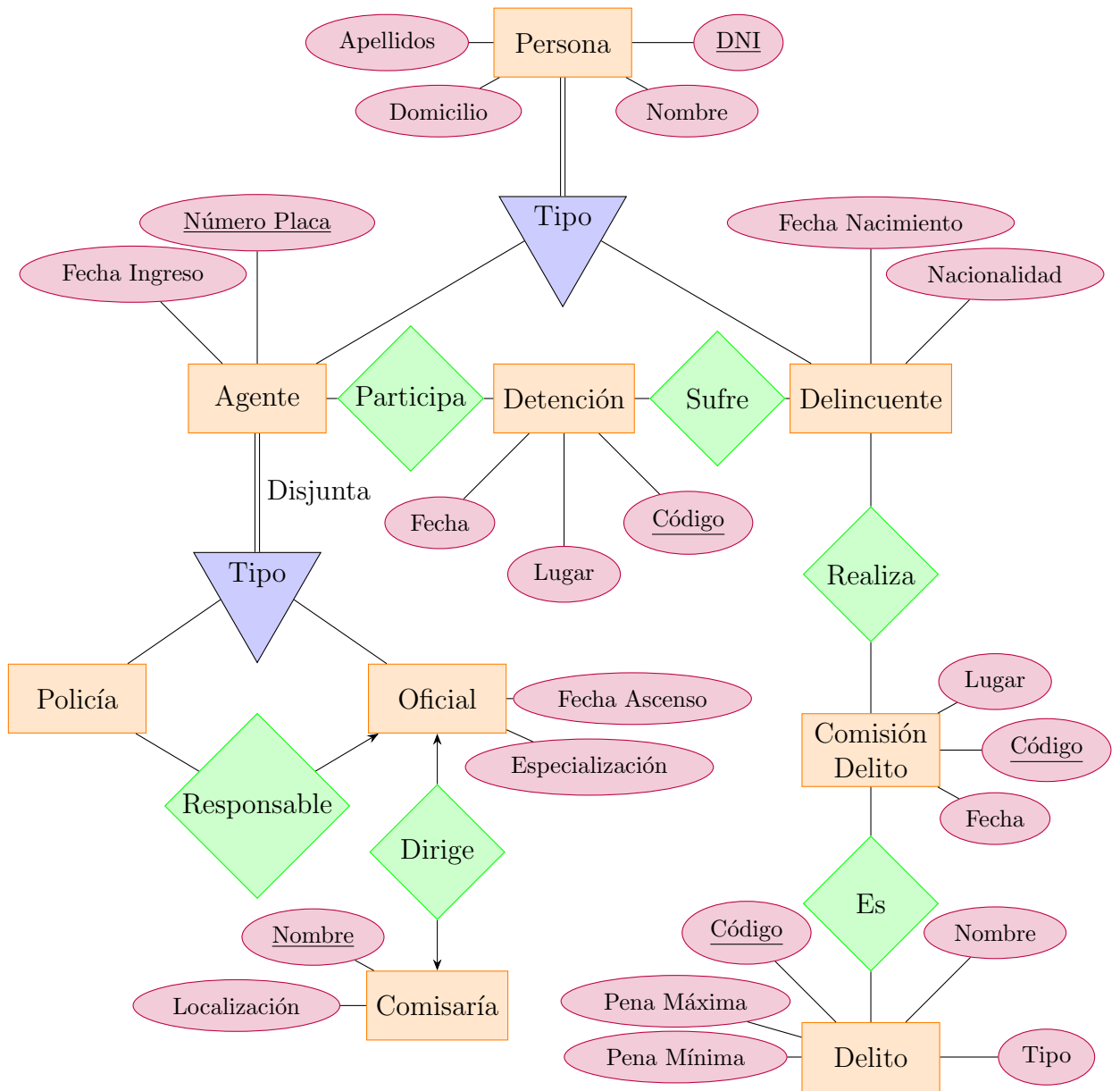


Figura 1.24: Diagrama Entidad-Relación del Ejercicio 1.1.12.

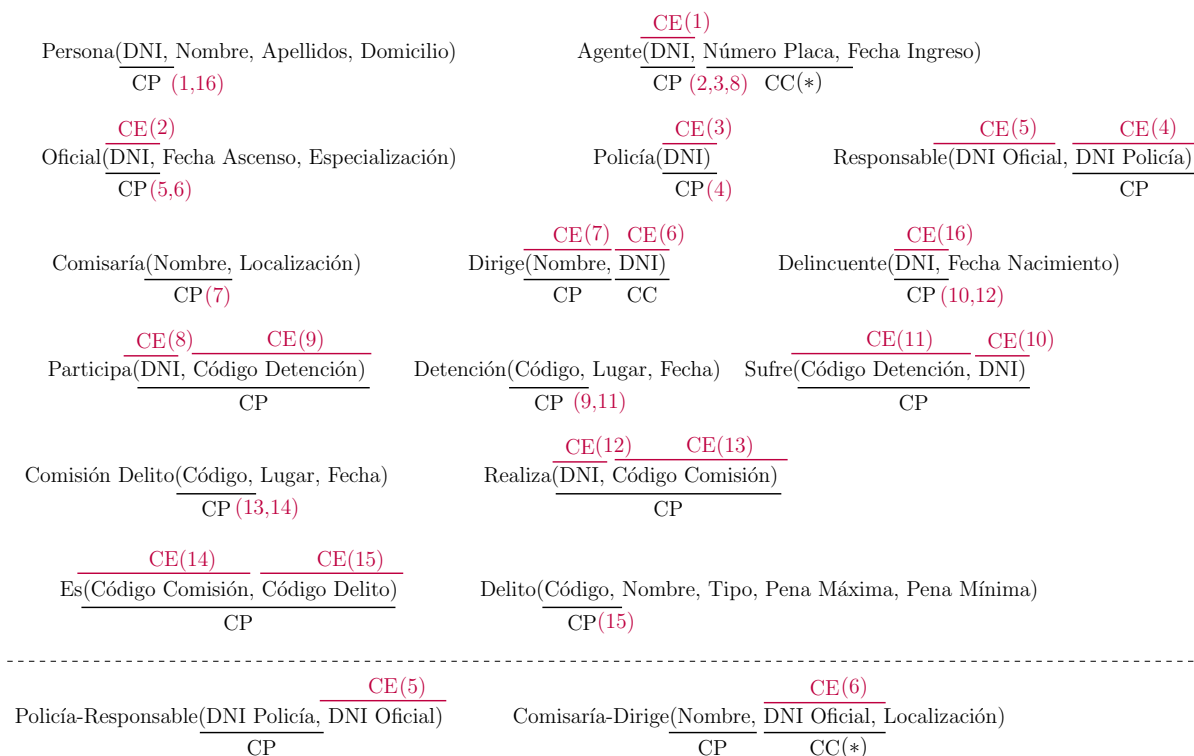


Figura 1.25: Tablas de Modelo E-R del Ejercicio 1.1.12.

- Objetos de baja emisión de luz: Planetas y Satélites.
- Objetos de alta emisión de luz: Estrellas.
- De los planetas almacenamos el tipo de planeta.
- De los satélites nos interesa saber el tipo de satélite.
- De las estrellas almacenamos el tipo y subtipo al que pertenecen.
- Además, queremos describir el hecho de que:
  - Un satélite gira alrededor de un planeta, conociéndose la distancia entre ellos, y también almacenamos a cuantos años luz se encuentran entre sí. Alrededor de un planeta pueden girar diferentes satélites.
  - Un planeta, junto con sus satélites, giran alrededor de una estrella. También se almacena a cuantos años luz distan entre sí. Alrededor de una estrella pueden girar diferentes planetas.
  - Un grupo de estrellas forman una constelación y cada estrella puede estar en varias constelaciones.
  - De las constelaciones nos interesa almacenar el código y nombre.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.26. Notemos que hemos supuesto que, además de planetas y satélites, también se pueden descubrir otros objetos astronómicos de baja emisión de luz.

El paso a tablas está en la Figura 1.27. La fusión de tablas se ha realizado por debajo de la línea divisoria discontinua.



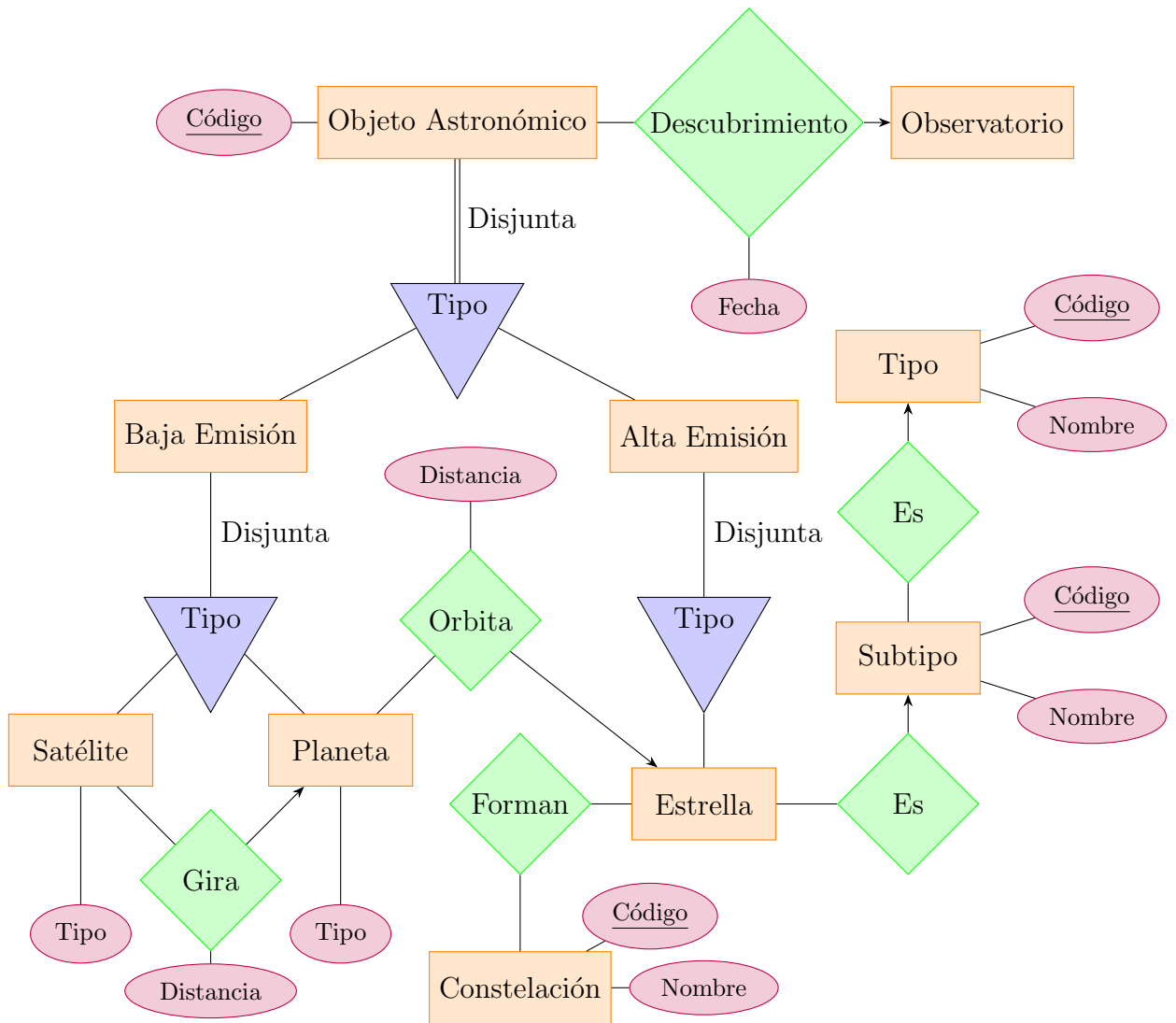


Figura 1.26: Diagrama Entidad-Relación del Ejercicio 1.1.13.

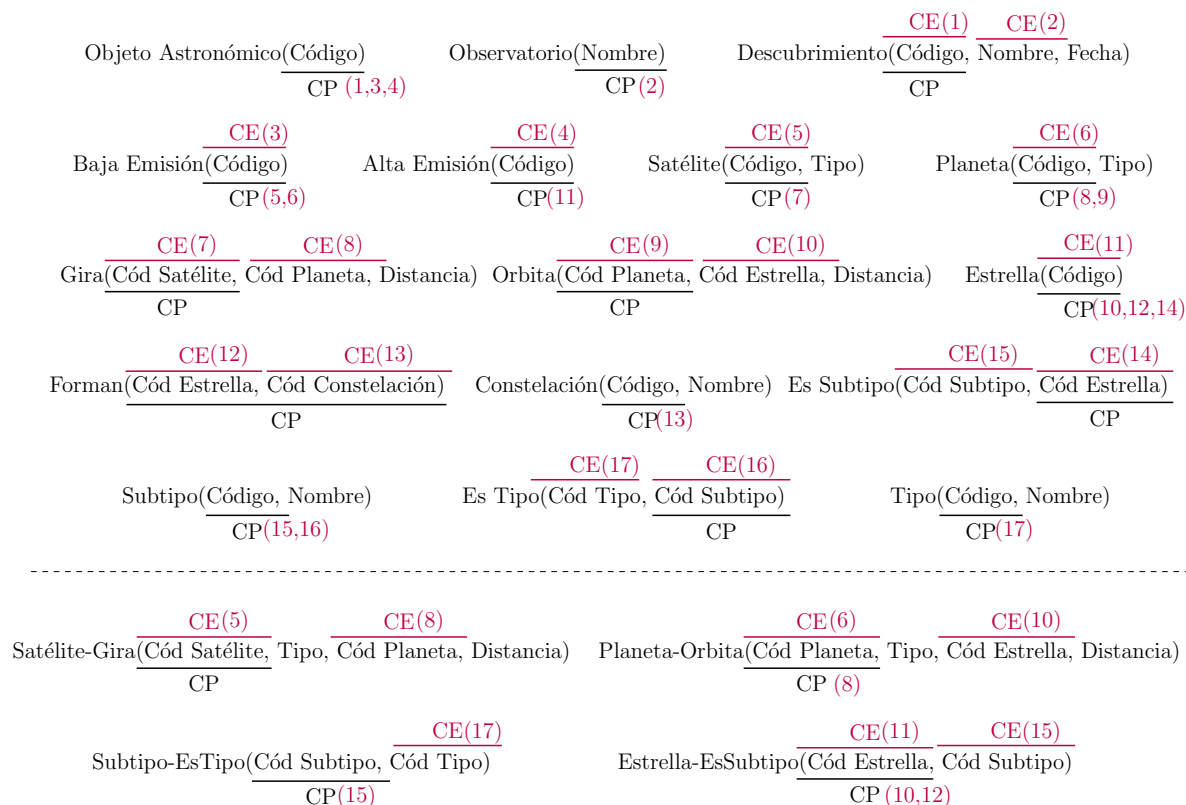


Figura 1.27: Tablas de Modelo E-R del Ejercicio 1.1.13.

**Ejercicio 1.1.14.** En una base de datos de un hospital hay información relativa a pacientes (de los que se almacena el DNI, nombre, dirección y teléfono) e historias clínicas (que tienen un código y una fecha de creación). Cada persona tiene una única historia clínica y cada historia clínica corresponde a una sola persona. Una historia clínica consta de distintos episodios o ingresos que se van numerando de forma consecutiva y de los cuales se registra además el motivo del episodio y la fecha en la que se produce.

El diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.28. Notemos que establecer la participación obligatoria de la relación *Tiene* en la entidad *Historia Clínica* podría ser válida.

El paso a tablas está en la Figura 1.29. La fusión de tablas se ha realizado por debajo de la línea divisoria discontinua.

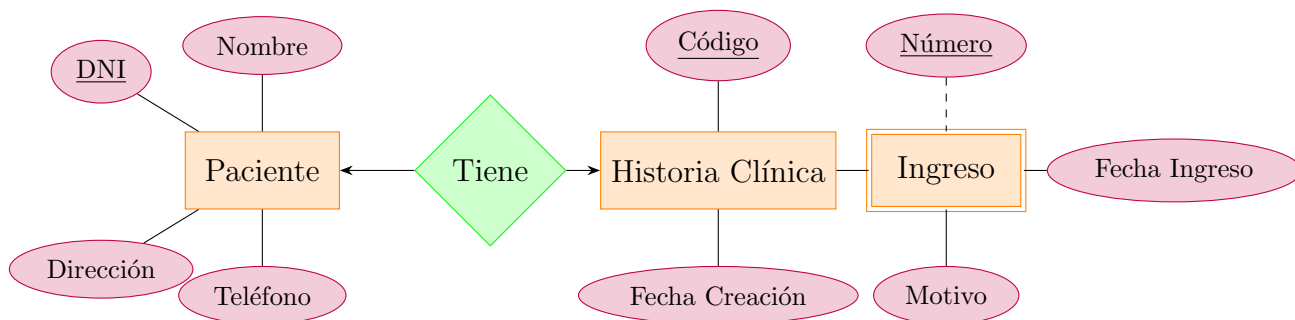


Figura 1.28: Diagrama Entidad-Relación del Ejercicio 1.1.14.

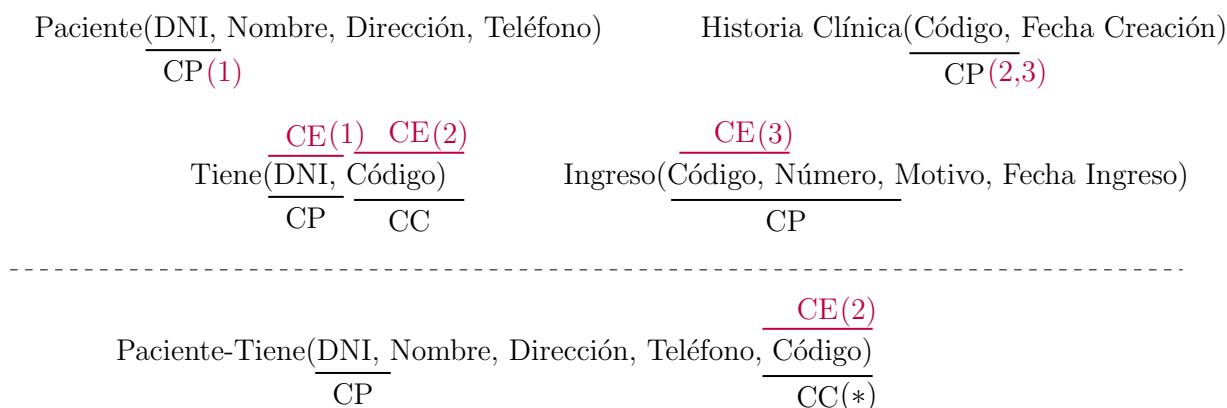


Figura 1.29: Tablas de Modelo E-R del Ejercicio 1.1.14.

**Ejercicio 1.1.15.** En una base de datos hay información relativa a empleados (sobre los que se conoce el identificador, nombre y dirección). Hay dos tipos de empleados: fijos y eventuales. De los fijos se conoce la fecha de inicio del contrato y la fecha de la última revisión salarial. De los eventuales, se conoce la fecha de contratación y la duración del contrato. En la empresa hay una serie de tareas (código y descripción). Determinadas tareas solo pueden hacerlas empleados fijos.

El diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.30.

El paso a tablas está en la Figura 1.31.

**Ejercicio 1.1.16.** En una base de datos hay información sobre proveedores (de los que se almacena el código de proveedor, nombre y dirección) y productos (de los que se almacena código del producto, nombre, precio y descripción). En general los proveedores suministran varios productos. Un producto puede ser suministrado por más de un proveedor. ¿Cómo cambia el diseño si se quiere registrar el precio al que cada proveedor suministra el producto?

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.32. Notemos que, en el caso de que se quiera registrar el precio al que cada proveedor suministra el producto, el atributo *Precio* debería ser parte de la relación *Suministra*, sin ser discriminante.

El paso a tablas está en la Figura 1.33. En la variante en la que se quiere registrar el precio al que cada proveedor suministra el producto, el atributo *Precio* debería ser parte de la relación *Suministra*, sin ser discriminante.

**Ejercicio 1.1.17.** En la base de datos hay información sobre el contenido de distintos libros de texto. Hay información relativa a los libros (de los que se almacena el ISBN, título y editorial) y de los capítulos de cada libro (que tienen un número de capítulo y un título de capítulo).

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.34.

El paso a tablas está en la Figura 1.35.

**Ejercicio 1.1.18.** Se desea diseñar una base de datos de restaurantes de la provincia de Granada. La información que se desea almacenar es la siguiente:

- De cada restaurante, almacenaremos nombre, CIF, domicilio. No hay dos restaurantes con el mismo nombre.

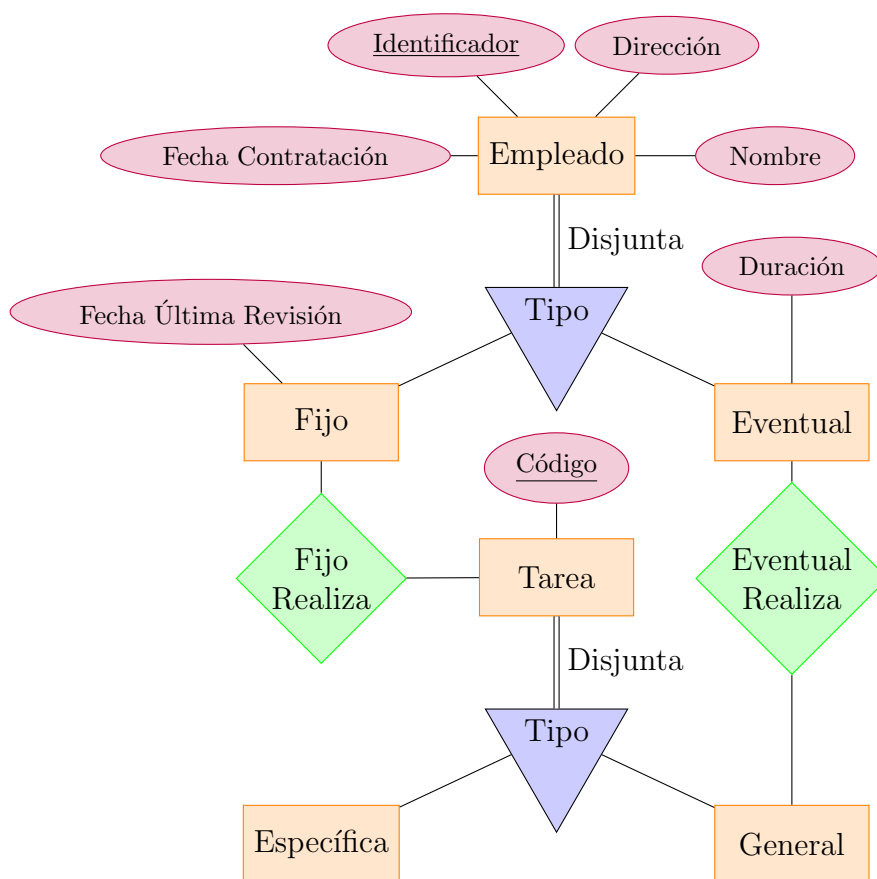


Figura 1.30: Diagrama Entidad-Relación del Ejercicio 1.1.15.

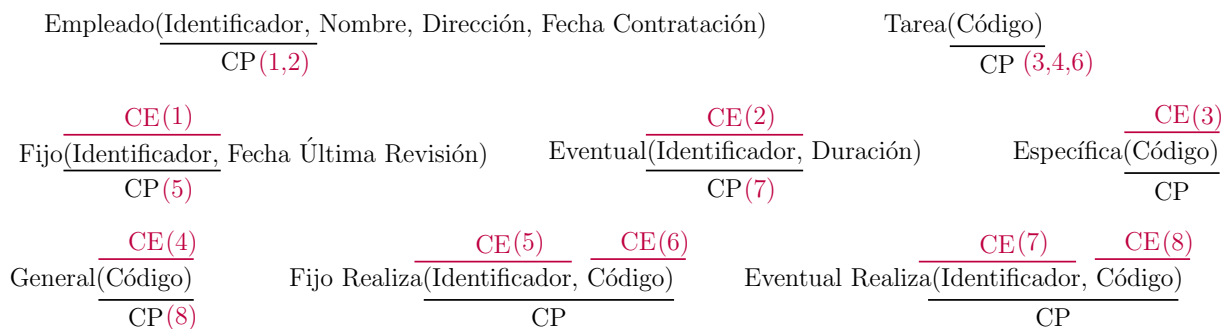


Figura 1.31: Tablas de Modelo E-R del Ejercicio 1.1.15.

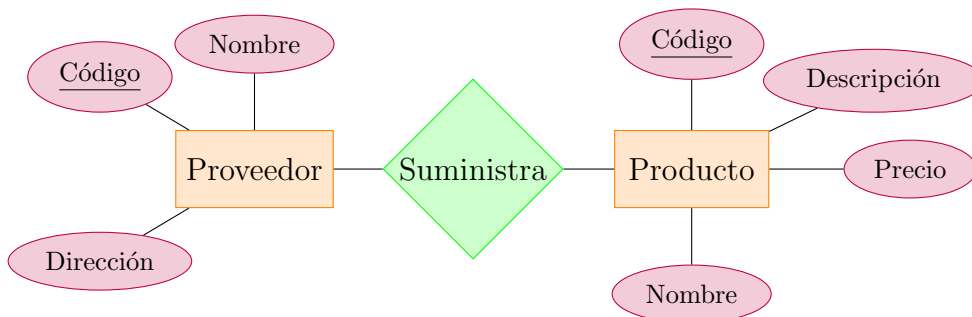


Figura 1.32: Diagrama Entidad-Relación del Ejercicio 1.1.16.

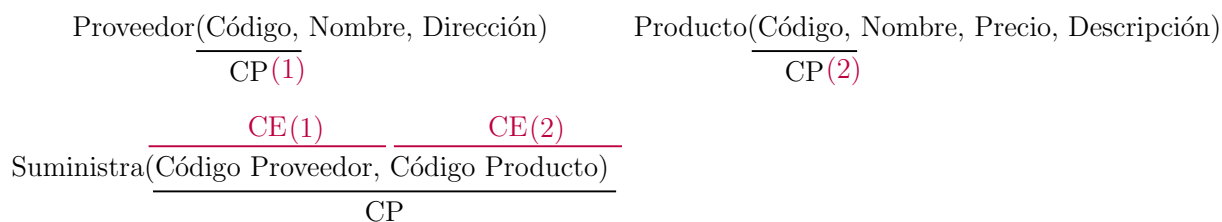


Figura 1.33: Tablas de Modelo E-R del Ejercicio 1.1.16.

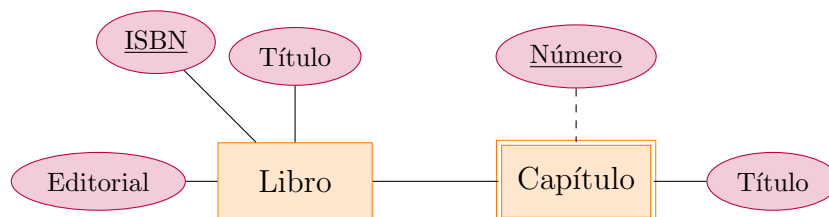


Figura 1.34: Diagrama Entidad-Relación del Ejercicio 1.1.17.

- Cada restaurante tiene unas personas propietarias (caracterizadas por su DNI, nombre y domicilio).
- De cada restaurante se desea registrar el horario de apertura, con la idea de poder facilitar luego información sobre qué restaurantes están abiertos en un momento dado.
- En cada restaurante trabaja un conjunto de empleados (nombre, DNI, domicilio). También se almacena el tipo de trabajo que desempeña el empleado en el restaurante, que puede variar de un restaurante a otro. Los propietarios de un restaurante también pueden ser empleados del mismo o de otro restaurante.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.36. Notemos que:

- No establecemos una relación de jerarquía, ya que el saber si alguien es propietario o empleado viene determinada por participar o no en cada relación.
- Hemos establecido el atributo *Tipo* como discriminante, ya que suponemos que un empleado puede trabajar en un restaurante varias veces con distintos trabajos.
- Respecto a los horarios, para poder consultar, por ejemplo, todos los restaurantes que abren un festivo a las 17h, hemos establecido una entidad débil *Tramo*, que representa cada tramo horario de apertura del restaurante. Mediante programación tendríamos que controlar que no se solapasen.



Figura 1.35: Tablas de Modelo E-R del Ejercicio 1.1.17.

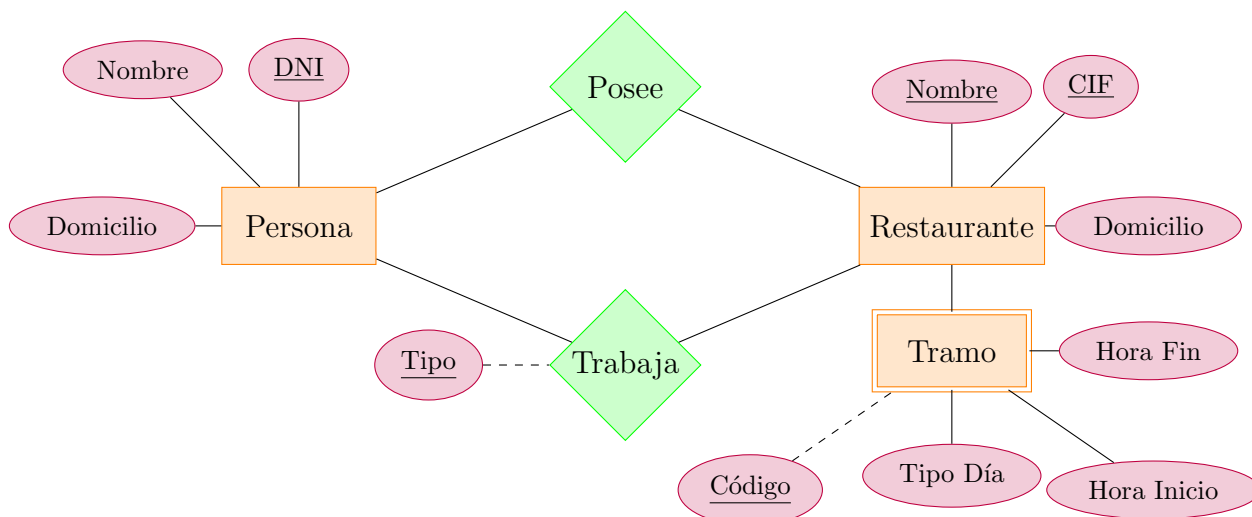


Figura 1.36: Diagrama Entidad-Relación del Ejercicio 1.1.18.

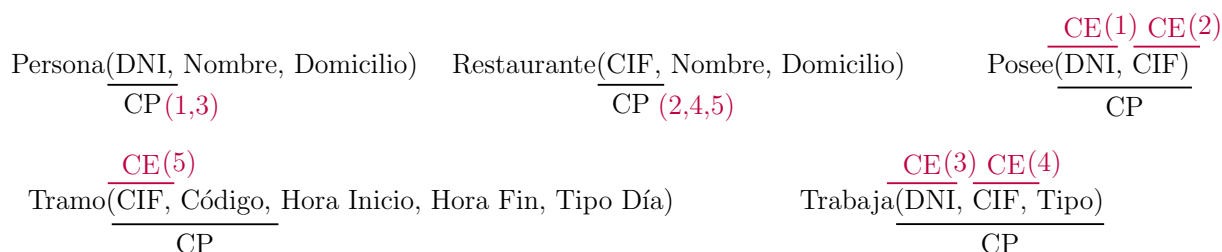


Figura 1.37: Tablas de Modelo E-R del Ejercicio 1.1.18.

El paso a tablas está en la Figura 1.37.

**Ejercicio 1.1.19.** Se desea elaborar una base de datos para almacenar datos de promociones inmobiliarias.

- De una promoción deseamos registrar su municipio, la dirección y el tipo de promoción (si es libre o de protección oficial).
- Una promoción se compone de una serie de viviendas que van numeradas y tienen un plano asociado.
- Las promociones tienen promotores. Hay dos tipos de promotores: privados (de los que se desea saber su NIF, nombre y años de experiencia) o públicos (de los que se desea saber su NIF, nombre y ámbito -municipal, autonómico, nacional). Los promotores públicos solo pueden llevar promociones de protección oficial.
- Asimismo, una promoción tiene asociado uno o varios constructores, de los que se desea registrar el NIF, el nombre y el tamaño de la empresa.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.38. El paso a tablas está en la Figura 1.39.

**Ejercicio 1.1.20.** Un estudio de arquitectura pretende diseñar una base de datos que le permita gestionar la información sobre la actividad de su negocio.

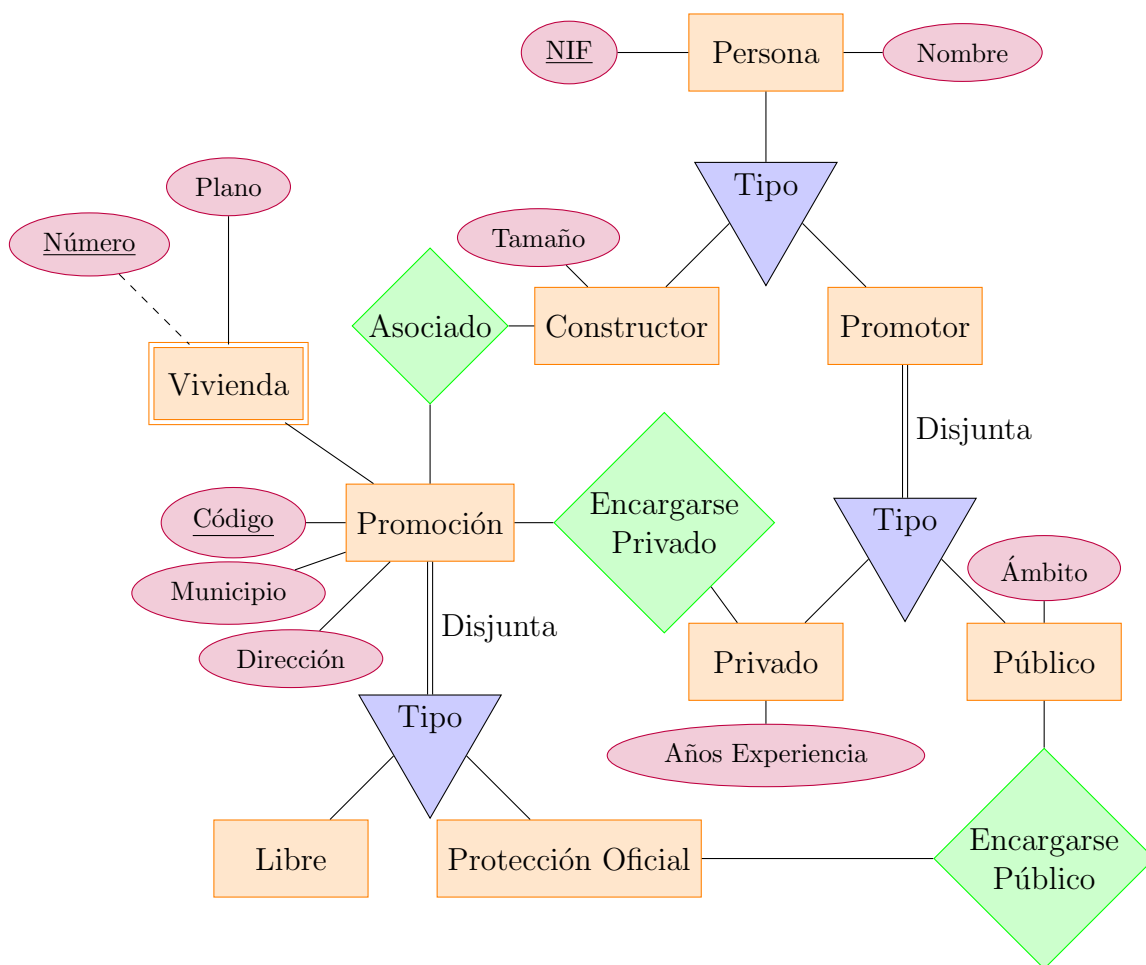


Figura 1.38: Diagrama Entidad-Relación del Ejercicio 1.1.19.

Persona( <u>NIF</u> , Nombre) CP(1,5)	Constructor( <u>NIF</u> , Tamaño) CP(2)	Asociado( <u>NIF</u> , Código) CP
Promoción( <u>Código</u> , Municipio, Dirección) CP(3,4,6,8,13)	Vivienda( <u>Código</u> , Número, Plano) CP	Promotor( <u>NIF</u> ) CP(11,12)
Protección Oficial( <u>Código</u> ) CP(10)	Encargarse Privado( <u>NIF</u> , Código) CP	Encargarse Público( <u>NIF</u> , Código) CP
Público( <u>NIF</u> , Ámbito) CP(9)	Privado( <u>NIF</u> , Años Experiencia) CP(7)	Libre( <u>Código</u> ) CP

Figura 1.39: Tablas de Modelo E-R del Ejercicio 1.1.19.

- En el estudio se realizan dos tipos de trabajo: proyectos de arquitectura y proyectos de interiorismo.
- Los proyectos de arquitectura constan de: código de proyecto, fecha del proyecto, presupuesto, varios planos (cada plano es un fichero), un documento de descripción de materiales y un documento con el permiso de obras.
- Los proyectos de interiorismo constan de: código de proyecto, fecha del proyecto, presupuesto, estilo y un documento de descripción del mobiliario.
- Los clientes del estudio pueden ser de dos tipos: clientes particulares y empresas.
- Para los clientes particulares se registra: número del cliente, DNI, nombre, dirección y varios teléfonos.
- Para las empresas se registra: número del cliente, CIF, nombre, dirección, teléfonos y observaciones sobre la forma de pago.
- Los empleados del estudio (de los que se conoce el número de empleado y nombre) tienen distintas categorías: Peritos, Arquitectos e Interioristas.
- Cada proyecto de arquitectura tiene asignado un arquitecto responsable y un conjunto de peritos que hacen el seguimiento de la obra. Cada proyecto de interiorismo tiene asignado un interiorista responsable.
- Cada cliente encarga la realización de uno o varios proyectos de igual o diferente tipo.

El Diagrama Entidad-Relación correspondiente se encuentra en la Figura 1.40.

El paso a tablas está en la Figura 1.41. La fusión de las tablas se ha realizado por debajo de la línea divisoria discontinua.

**Ejercicio 1.1.21.** Ponga un ejemplo original de restricciones semánticas y elabore el correspondiente esquema E/R para cada una de las siguientes situaciones:

- Una entidad fuerte y una débil. Variar el problema de forma que la entidad débil deje de serlo sin tocar la semántica de la relación que tiene con la entidad fuerte.
- Una jerarquía con obligatoriedad y exclusividad.
- Una jerarquía sin obligatoriedad, pero con exclusividad.
- Una jerarquía sin obligatoriedad ni exclusividad.
- Una jerarquía con obligatoriedad, pero sin exclusividad.
- Una relación uno a muchos, obligatoria para la parte de muchos.
- Una relación uno a uno obligatoria en uno de los sentidos.
- Una relación muchos a muchos con un atributo discriminador.



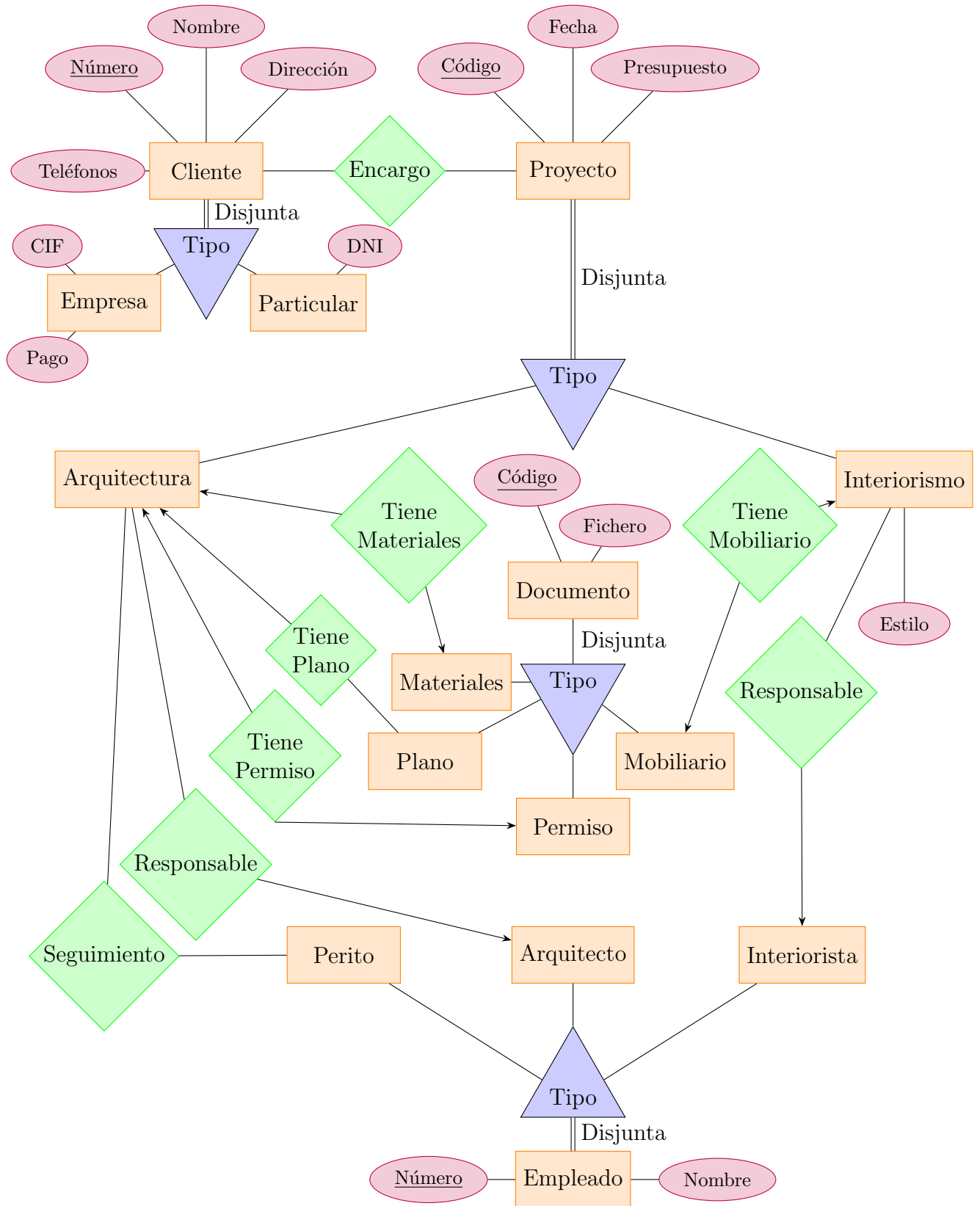


Figura 1.40: Diagrama Entidad-Relación del Ejercicio 1.1.20.

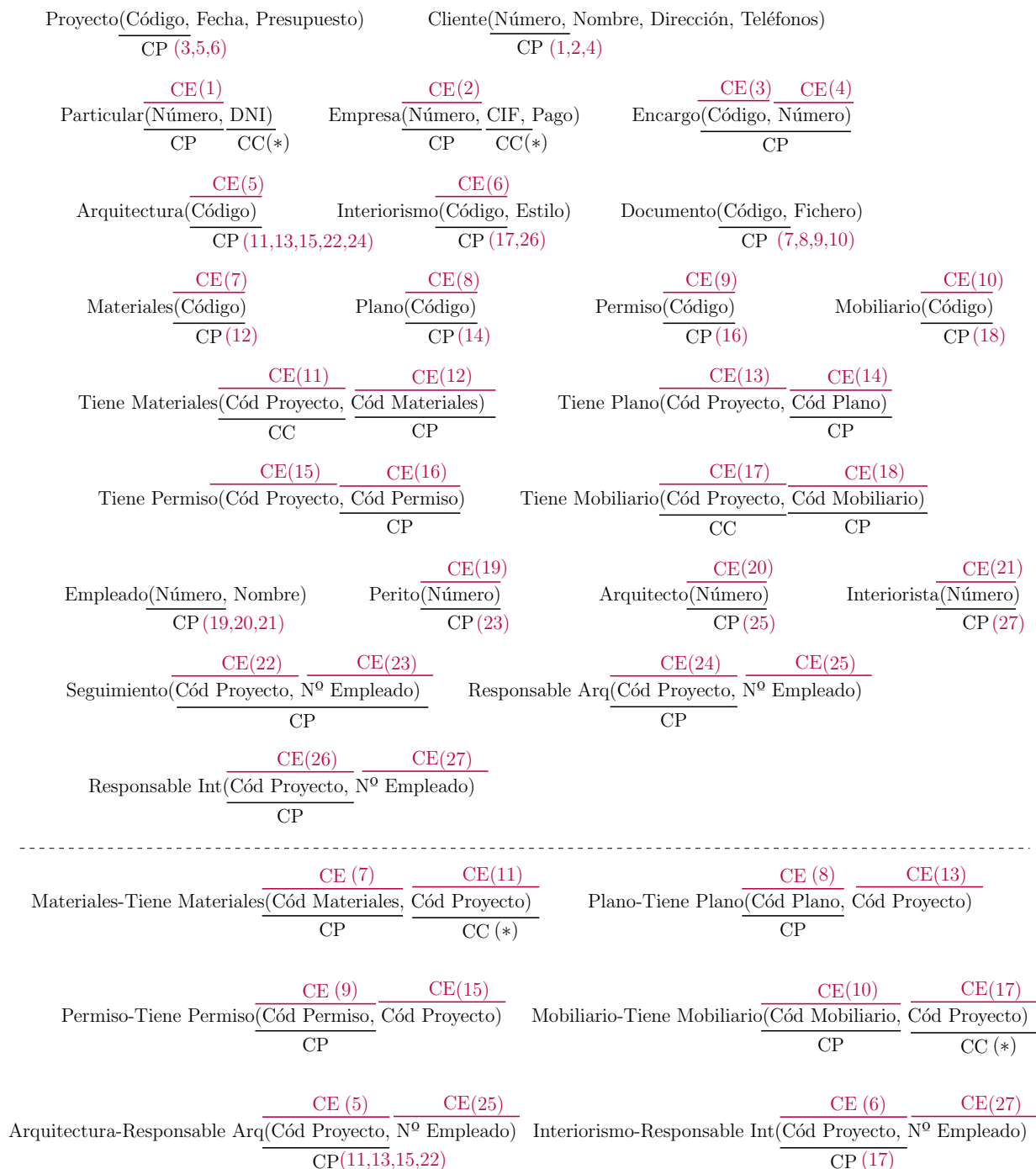


Figura 1.41: Tablas de Modelo E-R del Ejercicio 1.1.20.

- Una relación uno a uno con un atributo discriminador.
- Una agregación.
- Una relación ternaria.

## 1.2. Álgebra Relacional

**Ejercicio 1.2.1.** Considere el esquema de base de datos relacional de la Figura 1.42.

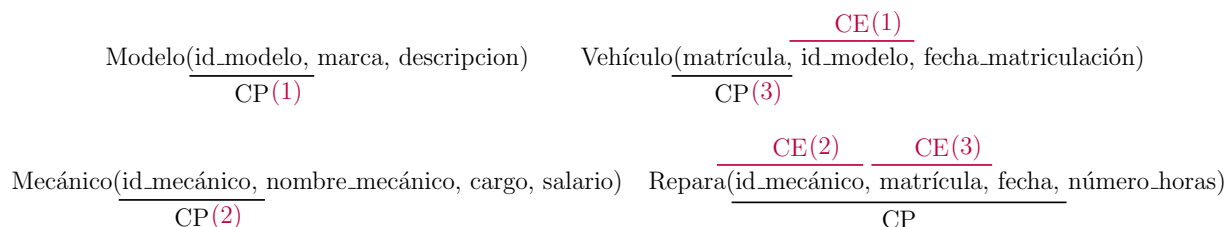


Figura 1.42: Esquema de base de datos relacional del Ejercicio 1.2.1.

Cada vehículo tiene asignado un modelo de una marca determinada. La tabla *Repara* registra reparaciones indicando qué mecánico repara qué vehículo en qué fecha y cuántas horas dura la reparación.

Indiqué qué consulta debe realizar en álgebra relacional para obtener la información solicitada en cada uno de los siguientes puntos:

1. Reparaciones de más de 20 horas.

$$\sigma_{\text{número\_horas} > 20}(\text{Repara})$$

2. Códigos de mecánicos que han reparado el vehículo de matrícula 1234ABC.

$$\pi_{\text{id\_mecánico}}(\sigma_{\text{matrícula}=1234ABC}(\text{Repara}))$$

3. Parejas de mecánicos que se pueden hacer en la empresa.

$$\begin{aligned} \rho(\text{Mecánico}) &= \text{Mecánico}_1, \text{Mecánico}_2 \\ \sigma_{\text{Mecánico}_1.\text{id\_mecánico} < \text{Mecánico}_2.\text{id\_mecánico}}(\text{Mecánico}_1 \times \text{Mecánico}_2) \end{aligned}$$

Notemos que, si tenemos dos mecánicos,  $A$  y  $B$ , debemos:

- Evitar las parejas de un mismo mecánico repetido, como  $A, A$  o  $B, B$ . Esto se consigue obligando a que los *id* sean distintos.
- Evitar parejas duplicadas, ya que  $A, B$  y  $B, A$  en realidad son distintas. Debido a que el orden en  $\mathbb{N}$  es un orden total<sup>1</sup>, esto se consigue obligando a que el código de uno de ellos (sea este el mecánico 1) sea menor que el del otro.

4. Marca de los vehículos matriculados después del 1/1/20.

$$\pi_{\text{marca}}(\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/20)}(\text{Vehículo}) \bowtie \text{Modelo})$$

<sup>1</sup>Suponemos que el dominio del código es  $\mathbb{N}$ . Si fuese alfanumérico, el orden alfabético también es total.

5. Parejas  $\langle \text{cargo}, \text{marca} \rangle$  entre las que se ha dado alguna reparación.

$$\pi_{\text{cargo,marca}}[\pi_{\text{id\_mecánico,cargo}}(\text{Mecánico}) \bowtie \pi_{\text{id\_mecánico,matrícula}}(\text{Repara}) \bowtie \\ \bowtie \pi_{\text{matrícula,id\_modelo}}(\text{Vehículo}) \bowtie \pi_{\text{id\_modelo,marca}}(\text{Modelo})]$$

Notemos que esto sería equivalente a la siguiente consulta, solo que anteriormente hemos ido reduciendo el número de tuplas a manejar:

$$\pi_{\text{cargo,marca}}(\text{Mecánico} \bowtie \text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})$$

6. Vehículos que o tienen una fecha de matriculación posterior al 1/1/22 o han sido reparados con posterioridad a esa misma fecha.

$$\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/22)}(\text{Vehículo}) \cup \\ \cup \pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}}(\text{Vehículo} \bowtie \sigma_{\text{fecha} > \text{fecha}(1/1/22)}(\text{Repara}))$$

7. Vehículos con fecha de matriculación posterior al 1/1/22 que han sido reparados alguna vez.

$$\pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}}(\sigma_{\text{fecha\_matriculación} > \text{fecha}(1/1/22)}(\text{Vehículo}) \bowtie \text{Repara})$$

8. Marca de los vehículos que no han tenido ninguna reparación en el año 2022.

$$\pi_{\text{marca}}[(\text{Vehículo} - \pi_{\substack{\text{matrícula} \\ \text{id\_modelo} \\ \text{fecha\_matriculación}}}(\text{Vehículo} \bowtie \\ \bowtie \sigma_{\text{fecha}(01/01/22) \leq \text{fecha} \leq \text{fecha}(31/12/22)}(\text{Repara}))) \bowtie \text{Modelo}]$$

9. Código de los mecánicos que han reparado vehículos de, al menos, dos marcas distintas.

Obtenemos en primer lugar todas las parejas de mecánicos con las marcas de los vehículos que han reparado; es decir:

$$\rho(\pi_{\text{id\_mecánico,marca}}(\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = A, B$$

La consulta sería:

$$\pi_{A.\text{id\_mecánico}} \left( A \bowtie_{\substack{A.\text{id\_mecánico}=B.\text{id\_mecánico} \\ A.\text{marca} \neq B.\text{marca}}} B \right)$$

10. Vehículos que tienen una sola reparación.

Buscamos en primer lugar los vehículos con más de una reparación. Para ello, tras hacer el producto cartesiano, exigimos que sea el mismo vehículo pero,

o bien haya sido reparado en fechas distintas, o bien haya sido reparado por mecánicos distintos.

$$\begin{aligned} \rho(\text{Repara}) &= A \\ \rho(\text{Repara}) &= B \\ \rho \left[ \pi_{\text{matrícula}} \left( A \bowtie \begin{array}{c} A.\text{matrícula}=B.\text{matrícula} \\ (A.\text{fecha}>B.\text{fecha} \vee A.\text{id\_mecánico}>B.\text{id\_mecánico}) \end{array} B \right) \right] &= C \end{aligned}$$

Notemos que tras proyectar habríamos llegado a la misma tabla estableciendo solo  $\neq$  en vez de  $>$ , pero usamos  $>$  ya que así obtenemos la mitad de las tuplas antes de proyectar.

Por tanto, en  $C$  tenemos las matrículas de los coches que tienen más de una reparación. Por tanto, las matrículas de los vehículos que tienen una sola reparación serían:

$$\pi_{\text{matrícula}}(\text{Repara}) - C$$

11. Vehículos que han sufrido las reparaciones con la duración más alta.

Este caso es obtener un máximo, algo que es muy común en álgebra relacional. Obtener el máximo no es fácil, pero podemos obtener los que *no* cumplen la condición de ser el máximo.

$$\begin{aligned} \rho(\text{Repara}) &= A, B \\ \rho [\sigma_{A.\text{número\_horas}<B.\text{número\_horas}}(A \times B)] &= C \\ \rho(\pi_{A.\text{número\_horas}}(C)) &= D \end{aligned}$$

Por tanto, en  $D$  obtenemos los valores de *número\_horas* para los que hay otro valor que es mayor; es decir, los que no son el máximo. Por tanto, los que son el máximo son:

$$\pi_{\text{número\_horas}}(\text{Repara}) - D$$

Con esa consulta tan solo hemos obtenido el máximo del atributo *número\_horas*, pero ahora necesitamos obtener los vehículos que corresponden con esas reparaciones. Estos son:

$$\pi_{\text{matrícula}}(\text{Repara} \bowtie (\pi_{\text{número\_horas}}(\text{Repara}) - D))$$

12. Mecánicos que tienen el salario más bajo.

En primer lugar, obtenemos los que no son candidatos a mínimo:

$$\begin{aligned} \rho(\text{Mecánico}) &= A, B \\ \rho [\sigma_{A.\text{salario}>B.\text{salario}}(A \times B)] &= C \\ \rho(\pi_{A.\text{salario}}(C)) &= D \end{aligned}$$

En  $D$ , obtenemos los salarios que no son el mínimo, ya que hay otro salario menor. Por tanto, el salario mínimo es:

$$\pi_{\text{salario}}(\text{Mecánico}) - D$$

Por tanto, los mecánicos que tienen el salario más bajo son:

$$\pi_{\text{id\_mecánico}}(\text{Mecánico} \bowtie (\pi_{\text{salario}}(\text{Mecánico}) - D))$$

13. Mecánicos cuyo salario es uno de los dos salarios más bajos.

Este apartado es similar al anterior, solo que hemos de repetir el procedimiento dos veces. Sea  $A$  la relación que contiene el resultado de la consulta anterior, es decir, los mecánicos con el salario más bajo. Consideramos ahora el resto de mecánicos:

$$\rho(\text{Mecánico} - (\text{Mecánico} \bowtie A)) = C, D$$

Por tanto, a esta nueva lista de mecánicos que no tienen el salario mínimo, le aplicamos el mismo procedimiento que en el apartado anterior para obtener los mecánicos con el segundo salario más bajo.

$$\begin{aligned} \rho[\sigma_{C.\text{salario} > D.\text{salario}}(C \times D)] &= E \\ \rho(\pi_{C.\text{salario}}(E)) &= F \end{aligned}$$

Por tanto, el segundo salario más bajo es:

$$\pi_{\text{salario}}(C) - F$$

Por tanto, los mecánicos cuyo salario es el segundo más bajo son:

$$\pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{salario}}(C) - F))$$

Por tanto, los mecánicos cuyo salario es uno de los dos salarios más bajos son:

$$A \cup \pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{salario}}(C) - F))$$

14. Vehículos que han sido reparados alguna vez por cada uno de los mecánicos.

Vemos que se trata de una consulta de división. Sean los siguientes alias:

$$\begin{aligned} \rho\left(\pi_{\substack{\text{matrícula} \\ \text{id\_mecánico}}}(\text{Repara})\right) &= R \\ \rho(\pi_{\text{id\_mecánico}}(\text{Mecánico})) &= S \end{aligned}$$

La consulta sería:

$$R \div S$$

Esta consulta devuelve una instancia  $w$  de la relación  $W[\text{matrícula}]$  que cumple que, para cada  $u \in w$ ,  $v \in s$  instancia de  $S$ , si  $r$  es una instancia de  $R$  se tiene:

$$\exists t \in r \mid t[\text{matrícula}] = u \wedge t[\text{id\_mecánico}] = v$$

Es decir, contiene las matrículas de los coches que han sido reparados por todos los mecánicos.

15. Mecánicos que han reparado vehículos de todas las marcas.

Obtenemos en primer lugar el par  $\langle \text{id\_mecánico}, \text{marca} \rangle$  de los vehículos que ha reparado cada mecánico.

$$\rho(\pi_{\text{id\_mecánico,marca}}(\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = R$$

El divisor sería la relación que contiene todas las marcas posibles.

$$\rho(\pi_{\text{marca}}(\text{Modelo})) = S$$

Por tanto, la consulta sería:

$$R \div S$$

16. Vehículos a los que el mecánico de id 123 les ha hecho todas las reparaciones.

De todas las reparaciones, no consideramos aquellas que han sido realizadas otro mecánico:

$$\pi_{\text{matrícula}}(\text{Repara}) - \pi_{\text{matrícula}}(\sigma_{\text{id\_mecánico} \neq 123}(\text{Repara}))$$

17. Marcas para las que todos sus vehículos han sido reparados alguna vez por un empleado con un salario superior a 30000.

$$\begin{aligned} \rho(\pi_{\text{matrícula}}(\text{Repara} \bowtie \sigma_{\text{salario} > 30000}(\text{Mecánico}))) &= A \\ \rho(\pi_{\text{matrícula}}(\text{Vehículo}) - A) &= B \end{aligned}$$

En  $B$  tenemos las matrículas de los coches que nunca han sido reparados por un mecánico con salario superior a 30000. Por tanto, nos interesa quedarnos con el resto de marcas, por lo que la consulta es:

$$\pi_{\text{marca}}(\text{Modelo}) - \pi_{\text{marca}}(\text{Modelo} \bowtie \text{Vehículo} \bowtie B)$$

*Observación.* En este caso, el enunciado puede llevar a pensar que se trata de una consulta de división, pero no es así porque no se trata de todos los vehículos, sino de todos los vehículos de una marca.

18. Vehículos que, para todos los cargos que hay en la empresa, han tenido al menos una reparación de más de 2 horas de duración con un empleado de ese cargo.

El dividendo  $R$  ha de ser los vehículos que han tenido al menos una reparación de más de 2 horas, con el cargo del mecánico que ha realizado la reparación.

$$\rho(\pi_{\text{matrícula,cargo}}(\sigma_{\text{número\_horas} > 2}(\text{Repara}) \bowtie \text{Mecánico})) = R$$

El divisor  $S$  sería la relación que contiene todos los cargos posibles.

$$\rho(\pi_{\text{cargo}}(\text{Mecánico})) = S$$

Por tanto, la consulta sería:

$$R \div S$$



19. Marcas para las que todos sus vehículos han sido reparados alguna vez por el mismo mecánico.

De nuevo, como no se trata de *todos los vehículos* sino de *todos los vehículos de una marca*, no se trata de una consulta directa de división. No obstante, esta consulta es compleja. Creamos los siguientes alias:

$$\begin{aligned}\rho(\pi_{\text{matrícula}}(\text{Vehículo})) &= V \\ \rho(\pi_{\text{id.mecánico}}(\text{Mecánico})) &= M_1 = M_2\end{aligned}$$

Obtenemos en primer lugar todas las parejas de mecánico con vehículo para los cuales no se ha realizado una reparación.

$$\rho\left((V \times M_1) - \pi_{\substack{\text{matrícula} \\ \text{id.mecánico}}}(\text{Repara})\right) = A$$

Obtenemos ahora todas las parejas marca, mecánico para los cuales este mecánico no ha reparado algún coche de dicha marca.

$$\pi_{\substack{\text{marca} \\ \text{id.mecánico}}}(A \bowtie \text{Vehículo} \bowtie \text{Modelo}) = B$$

Obtenemos ahora las marcas para los que todos los mecánicos no han reparado algún coche de dicha marca.

$$\rho(B \div M_1) = C$$

Por tanto, tenemos que las marcas para las que todos sus vehículos han sido reparados alguna vez por el mismo mecánico son:

$$\pi_{\text{marca}}(\text{Modelo}) - C$$

20. Mecánico más joven que ha reparado vehículos de todas las marcas.

*Observación.* Para esto, supondremos que la tabla *Mecánico* tiene un atributo *fecha\_nacimiento*.

Obtengamos los mecánicos que han reparado vehículos de todas las marcas.

$$\rho(\pi_{\text{id.mecánico,marca}}(\text{Repara} \bowtie \text{Vehículo} \bowtie \text{Modelo})) = R$$

El divisor sería la relación que contiene todas las marcas posibles.

$$\rho(\pi_{\text{marca}}(\text{Modelo})) = S$$

Por tanto, los mecánicos que han reparado vehículos de todas las marcas son:

$$\rho(\pi_{\text{id.mecánico}}(R \div S)) = C$$

Obtenemos ahora el mecánico más joven de entre estos. Para ello, obtenemos en primer lugar los mecánicos que no son candidatos a ser el más joven.

$$\begin{aligned}\rho(\text{Mecánico} \bowtie C) &= A, B \\ \rho[\sigma_{A.\text{fecha\_nacimiento} < B.\text{fecha\_nacimiento}}(A \times B)] &= D \\ \rho(\pi_{A.\text{fecha\_nacimiento}}(D)) &= E\end{aligned}$$

Por tanto, en  $E$  tenemos las fechas de nacimiento de los mecánicos de  $C$  que no son el más joven. Por tanto, la fecha de nacimiento del mecánico más joven es:

$$\pi_{\text{fecha\_nacimiento}}(C) - E$$

Por tanto, los mecánicos más jóvenes que han reparado vehículos de todas las marcas son:

$$\pi_{\text{id\_mecánico}}(C \bowtie (\pi_{\text{fecha\_nacimiento}}(C) - E))$$

**Ejercicio 1.2.2.** Considere el esquema de base de datos relacional de la Figura 1.43.

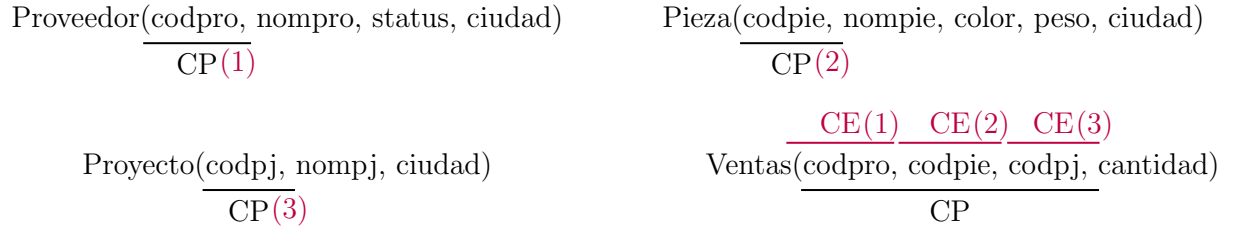


Figura 1.43: Esquema de base de datos relacional del Ejercicio 1.2.2.

Indique qué consulta debe realizar en álgebra relacional para obtener la información solicitada en cada uno de los siguientes puntos:

1. Encontrar todas las parejas de ciudades tales que la primera sea la de un proveedor y la segunda la de un proyecto entre los cuales haya algún suministro.

$$\pi_{\text{Proveedor.ciudad} \quad \text{Proyecto.ciudad}} \left[ \left( \pi_{\text{codpro} \quad \text{codpj}}(\text{Ventas}) \bowtie \text{Proveedor} \right) \bowtie_{\text{Ventas.codpj}=\text{Proyecto.codpj}} \text{Proyecto} \right]$$

Notemos que la segunda reunión realiza no puede ser natural, ya que ambas relaciones comparten el mismo nombre para un atributo *Ciudad*.

2. Encontrar los códigos de las piezas suministradas a algún proyecto por un proveedor que se encuentre en la misma ciudad que el proyecto.

$$\pi_{\text{codpie}} [(\text{Ventas} \bowtie \text{Proveedor}) \bowtie \text{Proyecto}]$$

Notemos que la segunda impone que la ciudad sea la misma, ya que se realiza una reunión natural y ambos atributos comparten nombre.

3. Encontrar los códigos de los proyectos que tienen al menos un proveedor que no se encuentre en su misma ciudad.

$$\pi_{\text{Proyecto.codpj}} \left[ (\text{Ventas} \bowtie \text{Proveedor}) \bowtie_{\substack{\text{Ventas.codpj}=\text{Proyecto.codpj} \\ \text{Proveedor.ciudad} \neq \text{Proyecto.ciudad}}} \text{Proyecto} \right]$$

4. Mostrar todas las ciudades de donde proceden piezas y las ciudades donde hay proyectos.

$$\pi_{\text{ciudad}}(\text{Pieza}) \cup \pi_{\text{ciudad}}(\text{Proyecto})$$

5. Mostrar todas las ciudades de los proveedores en las que no se fabriquen piezas.

$$\pi_{\text{ciudad}}(\text{Proveedor}) - \pi_{\text{ciudad}}(\text{Pieza})$$

6. Mostrar todas las ciudades de los proveedores en las que además se fabriquen piezas.

$$\pi_{\text{ciudad}}(\text{Proveedor}) \cap \pi_{\text{ciudad}}(\text{Pieza})$$

7. Encontrar los códigos de los proyectos que usan una pieza que vende S1.

$$\pi_{\text{codpj}} (\sigma_{\text{codpro}=\text{S1}}(\text{Ventas}))$$

8. Encontrar la cantidad más pequeña enviada en algún suministro.

Se trata de encontrar el mínimo del atributo *cantidad* de la relación *Ventas*.

$$\begin{aligned}\rho(\text{Ventas}) &= A, B \\ \rho[\sigma_{A.cantidad < B.cantidad}(A \times B)] &= C \\ \rho(\pi_{B.cantidad}(C)) &= D\end{aligned}$$

Por tanto, la cantidad más pequeña enviada en algún suministro es:

$$\pi_{\text{cantidad}}(\text{Ventas}) - D$$

9. Encontrar los códigos de los proyectos que no utilizan una pieza roja suministrada por un proveedor de Londres.

$$\begin{aligned}\rho[\pi_{\text{codpj}}((\text{Ventas} \bowtie \sigma_{\text{ciudad}=\text{Londres}}(\text{Proveedor})) \\ \bowtie_{\text{Ventas.codpie}=\text{Pieza.codpie}} \sigma_{\text{Pieza.color}=\text{rojo}}(\text{Pieza}))] &= A \\ \pi_{\text{codpj}}(\text{Proyecto}) - A\end{aligned}$$

10. Encontrar los códigos de los proyectos que tienen como único proveedor a S1.

$$\pi_{\text{codpj}}(\text{Ventas}) - \pi_{\text{codpj}}(\sigma_{\text{codpro} \neq \text{S1}}(\text{Ventas}))$$

11. Encontrar los códigos de las piezas que se suministran a todos los proyectos de París.

Se trata de una división. El dividendo es:

$$\rho \left[ \begin{array}{c} \pi_{\text{codpie}} \\ \text{codpj} \end{array} (\text{Ventas}) \right] = A$$

El divisor es:

$$\rho[\pi_{\text{codpj}}(\sigma_{\text{ciudad}=\text{París}}(\text{Proyecto}))] = B$$

Por tanto, la consulta sería:

$$A \div B$$

12. Encontrar los códigos de los proveedores que venden la misma pieza a todos los proyectos.

El dividendo es:

$$\rho \left[ \begin{array}{c} \pi_{\text{codpro}} \\ \text{codpie} \\ \text{codpj} \end{array} (\text{Ventas}) \right] = A$$

El divisor es:

$$\rho[\pi_{\text{codpj}}(\text{Proyecto})] = B$$

La consulta es:

$$\pi_{\text{codpro}}(A \div B)$$

13. Encontrar los códigos de los proyectos a los que el proveedor S1 suministra todas las piezas existentes.

El dividendo es:

$$\rho \left[ \pi_{\substack{\text{codpj} \\ \text{codpie}}} (\sigma_{\text{codpro}=\text{S1}}(\text{Ventas})) \right] = A$$

El divisor es:

$$\rho [\pi_{\text{codpie}}(\text{Pieza})] = B$$

La consulta es:

$$A \div B$$

donde notemos que no es necesario proyectar sobre el código del proyecto, ya que la división ya lo hace.

14. Mostrar los códigos de los proveedores que suministran todas las piezas a todos los proyectos.

Podríamos pensar en hacer dos divisiones. Veamos este proceso:

$$\begin{aligned} \rho \left[ \pi_{\substack{\text{codpro} \\ \text{codpie} \\ \text{codpj}}} (\text{Ventas}) \right] &= A \\ \rho [\pi_{\text{codpie}}(\text{Pieza})] &= B \\ \rho [\pi_{\text{codpj}}(\text{Proyecto})] &= C \end{aligned}$$

Podríamos pensar entonces que la consulta sería:

$$\pi_{\text{codpro}}((A \div B) \div C)$$

No obstante, aquí obtenemos los códigos que proporcionan todas las piezas y que proporcionan a todos los proyectos, pero no necesariamente todas las piezas a todos los proyectos. Para esto último, hemos de considerar todas las parejas posibles de piezas y proyectos, y obtener los códigos de los proveedores que suministran todas las parejas posibles.

$$\begin{aligned} \rho \left[ \pi_{\substack{\text{codpro} \\ \text{codpie} \\ \text{codpj}}} (\text{Ventas}) \right] &= A \\ \rho [\pi_{\text{codpie}}(\text{Pieza}) \times \pi_{\text{codpj}}(\text{Proyecto})] &= B \end{aligned}$$

La consulta por tanto queda:

$$A \div B$$

15. Pieza con más peso entre las que pesan menos de 100.

Partimos de las piezas que pesan menos de 100:

$$\rho [\sigma_{\text{peso}<100}(\text{Pieza})] = A, B$$

Ahora, obtenemos los que no son candidatos a máximo de entre estas piezas:

$$\begin{aligned}\rho[\sigma_{A.peso < B.peso}(A \times B)] &= C \\ \rho[\pi_{A.peso}(C)] &= D\end{aligned}$$

Por tanto, el mayor peso menor de 100 es:

$$\rho[\pi_{\text{peso}}(A) - D] = E$$

Por tanto, la pieza con más peso entre las que pesan menos de 100 es:

$$\pi_{\text{codpie}}[A \bowtie E]$$

16. Entre los proyectos de Jaén, mostrar al que le han suministrado la pieza de mayor peso (puede haber más de uno).

Obtenemos los proyectos de Jaén:

$$\rho[\sigma_{\text{ciudad}=\text{Jaén}}(\text{Proyecto})] = A$$

Obtenemos ahora los pares de proyecto de Jaén, peso de una pieza que se ha suministrado a estos proyectos:

$$\rho\left[\pi_{\text{codpj}}^{\text{peso}}\left(\left(A \bowtie \text{Ventas}\right) \bowtie_{\text{ventas.codpie}=\text{pieza.codpie}} \text{Pieza}\right)\right] = B, C$$

Buscamos ahora los que no son candidatos a máximo:

$$\begin{aligned}\rho[\sigma_{B.peso < C.peso}(B \times C)] &= D \\ \rho[\pi_{B.peso}(D)] &= E\end{aligned}$$

Por tanto, el mayor peso suministrado a los proyectos de Jaén es:

$$\rho[\pi_{\text{peso}}(B) - E] = F$$

Por tanto, los proyectos de Jaén a los que le han suministrado la pieza de mayor peso son:

$$\pi_{\text{codpj}}[B \bowtie F]$$

17. Proyectos para los que la lista de piezas que les han suministrado tiene al menos dos piezas distintas.

Obtenemos en primer lugar las parejas de proyecto, pieza que se les ha suministrado:

$$\rho\left[\pi_{\text{codpj}}^{\text{codpie}}(\text{Ventas})\right] = A, B$$

Obtenemos ahora los proyectos para los que la lista de piezas que les han suministrado tiene al menos dos piezas distintas:

$$\pi_{A.\text{codpj}}\left[\sigma_{\substack{A.\text{codpj}=B.\text{codpj} \\ A.\text{codpie} \neq B.\text{codpie}}}(A \times B)\right]$$

18. Proyectos para los que la lista de piezas que les han suministrado tiene exactamente dos piezas distintas.

Sea  $D$  la relación que contiene los códigos de proyectos que tienen al menos dos piezas distintas (resuelto en el apartado anterior). Obtenemos ahora las parejas de proyecto, pieza que se les ha suministrado:

$$\rho \left[ \pi_{\substack{\text{codpj} \\ \text{codpie}}} (\text{Ventas}) \right] = A, B, C$$

Los proyectos a los que se les han suministrado al menos tres piezas distintas son:

$$\rho \left[ \pi_{A.\text{codpj}} \left[ \sigma_{\substack{A.\text{codpj}=B.\text{codpj} \\ B.\text{codpj}=C.\text{codpj} \\ A.\text{codpie} \neq B.\text{codpie} \\ A.\text{codpie} \neq C.\text{codpie} \\ B.\text{codpie} \neq C.\text{codpie}} (A \times B \times C) \right] \right] = E$$

Notemos que la eficiencia de esta consulta no es la mejor, pero podríamos hacer la selección de  $A \times B$  y luego hacer el producto con  $C$ . Por tanto, como los que han sido suministrados al menos dos piezas distintas son los que han suministrado al menos dos piezas distintas y no han suministrado al menos tres piezas distintas, la consulta sería:

$$D - E$$

19. Proveedores que han hecho una o dos ventas (y no más).

Buscaremos en primer lugar los proveedores que han hecho al menos tres ventas. Consideramos las tripletas de proveedor, pieza, proyecto que se han vendido:

$$\rho \left[ \pi_{\substack{\text{codpro} \\ \text{codpie} \\ \text{codpj}}} (\text{Ventas}) \right] = A, B, C$$

Para que el proveedor sea el mismo, una de las condiciones que debemos imponer es:

$$\theta_1 = A.\text{codpro} = B.\text{codpro} \wedge A.\text{codpro} = C.\text{codpro}$$

Para que dos ventas sean distintas teniendo el mismo proveedor, o bien la pieza o bien el proyecto han de ser distintos:

$$\begin{aligned} \theta_{AB} &= A.\text{codpie} \neq B.\text{codpie} \vee A.\text{codpj} \neq B.\text{codpj} \\ \theta_{AC} &= A.\text{codpie} \neq C.\text{codpie} \vee A.\text{codpj} \neq C.\text{codpj} \\ \theta_{BC} &= B.\text{codpie} \neq C.\text{codpie} \vee B.\text{codpj} \neq C.\text{codpj} \end{aligned}$$

Por tanto, los proveedores que han hecho al menos tres ventas son:

$$\rho [\pi_{A.\text{codpro}} [\sigma_{\theta_1 \wedge \theta_{AB} \wedge \theta_{AC} \wedge \theta_{BC}} (A \times B \times C)]] = D$$

Por tanto, los proveedores que han hecho una o dos ventas son:

$$\pi_{\text{codpro}}(\text{Ventas}) - D$$

20. Proveedores en los que todos sus suministros son de una pieza roja o de una pieza de Granada.

$$\pi_{\text{codpro}}(\text{Ventas}) - \pi_{\text{codpro}}(\text{Ventas} \bowtie \sigma_{\text{color} \neq \text{rojo} \wedge \text{ciudad} \neq \text{Granada}}(\text{Piezas}))$$