



## Bases de datos relacionales

### Bases de datos

Customer

Customer_id
Firstname
Lastname
Address
Postal_code
Age
Gender
Email
Order_id
Invoice_id

#### Product

Product_id
Product_name
Amount
Price
Description
Image
Date_time
Status
Statistic

#### Order

Order_id
Total
Product_id
Customer_id
Date_time
Remark

# Índice



## 2.1. Modelización conceptual del software

- 2.1.1. Modelización de datos
- 2.1.2. Diccionario de datos
- 2.1.3. Modelo conceptual de datos (MCD)

## 2.2. Diagrama entidad/relación

- 2.2.1. Entidad
- 2.2.2. Relación
- 2.2.3. Cardinalidad y modalidad
- 2.2.4. Atributos de relación
- 2.2.5. Clave primaria y claves candidatas
- 2.2.6. Ejemplo de diagrama entidad/relación

## 2.3. Diagrama entidad/relación extendido



# Introducción

Los primeros atisbos del diseño relacional de bases de datos se dan en los años 60 y 70 por parte del investigador de IBM Edgar Frank Codd. Este publicará un trabajo al que da el nombre de "un modelo relacional de datos para grandes bancos de datos compartidos"; donde propone una serie de instrucciones lógicas y algebraicas que nos ayudan a poder gestionar de manera eficiente la gestión e la información dentro de las bases de datos.

Este modelo tiene su principal característica en las relaciones que se establezcan entre los datos, los que nos definen la estructura lógica necesaria para entender de qué manera se encuentra almacenada la información. Además, el modelo relacional nos hace ver la base de datos como una serie de relaciones que se manipulan usando un lenguaje relacional estándar. Este lenguaje es el llamado SQL (Structured Query Language) que se basa en la realización de consultas y se encuentra implementado en los sistemas de bases de datos más importantes, los llamados SGBD (Sistemas Gestores de Bases de Datos).

Los SGBD nos ayudan a que haya una comunicación entre la base de datos, el usuario, y las aplicaciones que necesiten del uso de esta. Además, estos sistemas proporcionan distintos beneficios en cuanto

a seguridad, integridad de los datos y manejo de funcionalidades se refiere.

Los objetivos que Codd escribe en su trabajo son los siguientes:

- > **Independencia lógica y física de los datos.** Cada vez que haya un cambio físico de almacenamiento las aplicaciones y usuarios deben de notar total transparencia. Así mismo, los cambios en los datos no implicarán un cambio en la estructura de las aplicaciones.
- > **Flexibilidad y seguridad.** Dependiendo de que usuario y con que aplicación se recupere la información almacenada, se mostrará de distintas maneras.
- > **Estandarización lógica.** Organizaremos los datos en tablas que entre ellas comparten la misma filosofía de creación y estarán relacionadas entre ellas también.
- > **Simplicidad.** Este modelo pretende que haya una mayor simplicidad a la hora del manejo de los datos frente a las demás opciones además de usar un lenguaje lo más natural dentro de lo que cabe.

## Al finalizar esta unidad

- + Ubicaremos el diseño de bases de datos relacionales en el marco del análisis de desarrollo del software.
- + Nos familiarizaremos con el concepto de diccionario de datos.
- + Comprenderemos la utilidad del diagrama entidad/relación como herramienta primordial en el diseño de bases de datos relacionales.
- + Dominaremos los conceptos y la terminología asociados al diagrama entidad/relación.
- + Describiremos apropiadamente la cardinalidad y modalidad de una relación.
- + Conoceremos los elementos que definen el diagrama entidad/relación extendida.



# 2.1.

## Modelización conceptual del software

Todos los sistemas de información parten de un problema a solucionar. Para poder solucionar esta problemática es necesario que se realice un examen de las necesidades que tenemos que cubrir.

Roger S. Pressman propuso una metodología para efectuar dicho análisis. En este análisis se toma el diccionario de datos como centro de arquitectura y se centra en el contexto relacional, dividiendo la problemática en tres distintos ámbitos representados mediante herramientas gráficas:

- > **Descripción de objetos de datos:** se representan los objetos de datos y sus relaciones. Se usa el diagrama de entidad/relación (DER).
- > **Especificación de proceso:** identificamos todas las funciones que debe desarrollar el sistema y como se relacionan entre ellas además de la forma en la que la información va a ser transformada cuando pase por dichas funciones. Esto se expresa con el diagrama de flujo de datos (DFD).
- > **Especificación de control:** va mostrando los estados en los que el sistema se va a encontrar a medida que avanzamos en el proceso, así como las transiciones que hacen que estos estados cambien basándose en el diagrama de transición de datos (DTE).

En el temario de base de datos y dado a que es la problemática que nosotros tenemos, nos vamos a centrar en el primero.

### 2.1.1. Modelización de datos

A la hora del proceso de desarrollo del sistema de información se irán modelando los datos en varias fases:

1. Creamos el modelo conceptual de datos nada más comenzar con el proceso de análisis.
2. Damos forma a este modelo con el diagrama entidad/relación, conformando un modelo lógico de datos.
3. Cuando comencemos a diseñar el software, iremos perfeccionando el diagrama entidad/relación, pasando a ser un modelo físico de datos.
4. De manera final, implantaremos este modelo en nuestro sistema gestor de bases de datos.

### 2.1.2. Diccionario de datos

El diccionario de datos o metabase se encarga de almacenar los datos sobre los datos, los llamados metadatos. El diccionario de datos es una herramienta de uso recurrente para un desarrollador de software a la hora de encontrar información sobre distintos elementos de una base de datos. En este diccionario se categorizan los datos de manera lógica, incluyendo descripciones, significado, estructura y consideraciones de edición, uso y seguridad de dichos datos.



...

**Tabla:** TDatos

**Campos:**

cDNI	CHAR(9)	NOT NULL UNIQUE
cNombre	VARCHAR(30)	NOT NULL
cApellidos	VARCHAR(60)	NOT NULL
cDirección	VARCHAR(100)	
cTeléfono	CHAR(9)	NOT NULL
dNacimiento	DATE	NOT NULL

**Clave primaria:** cDNI

**Claves ajenas:**

N/A

**Índices:**

iDatos_PK	cDNI ASC	
iDatos_Apellidos_Nombre		cApellidos ASC + cNombre ASC
iDatos_Nacimiento	dNacimiento DESC	

**Tabla:** TPrueba

**Campos:**

cFirma	VARCHAR(15)	NOT NULL
cDNI	CHAR(9)	NOT NULL
dFecha	DATE	NOT NULL

**Clave primaria:** cFirma ASC + cDNI ASC + dFecha DESC

**Claves ajenas:**

cFirma	TEjemplo.cSignatura
cDNI	TDatos.cNIF

**Índices:**

iPrueba_PK	cFirma ASC + cDNI ASC + dFecha DESC
iPrueba_FK_DNI	cDNI ASC
iPrueba_FK_Firma	cFirma ASC

...

Imagen 1. Ejemplo de un fragmento de un diccionario de datos.

### 2.1.3. Modelo conceptual de datos (MCD)

En este modelo se representa una visión más estática y estructurada de los datos, identificando la estructura interna de datos y las relaciones que existen entre las propias entidades.

**Características del MCD:**

- > Tiene que contener toda la información que se maneja en el sistema.
- > El estado final de los datos debe de estar representado.
- > Todos los cambios que se realicen en el sistema de información deben de ir reflejados en el modelo de datos, y a su vez, al revés también.

Toda esta modelización conceptual de datos se lleva a cabo en la gran mayoría de las ocasiones mediante un diagrama entidad/relación, que veremos más adelante.





# 2.2.

## Diagrama entidad/relación

El diagrama entidad/relación es, según Métrica-3:

"Una técnica cuyo objetivo es la representación y definición de todos los datos que se introducen, almacenan, transforman y producen dentro de un sistema de información, sin tener en cuenta las necesidades de la tecnología existente, ni otras restricciones."

Es necesario que se recalque la importancia de la independencia respecto al resultado e implementación que se nos dé al final. Este diagrama nos servirá para obtener la solución al problema que se plantea de manera indiferente a que SGBD se vaya a usar.

### 2.2.1. Entidad

Una entidad es un concepto abstracto que hace referencia a un objeto real o abstracto. Su existencia depende solo de sí misma y es necesario que su identificación sea clara y precisa. Se trata del elemento principal del modelo entidad/relación y se nombran con sustantivos en singular que encierra un concepto que el analista debe de identificar. Un ejemplo puede ser: "Trabajador".

Las entidades contienen elementos concretos referentes a sí mismas, las *ocurrencias*. Por ejemplo, cada trabajador en específico es una ocurrencia del ejemplo de entidad anterior "Trabajador".

Pero dentro de cada ocurrencia, para poder distinguirlas, habrá que tener otros datos asociados (nombre, dirección, DNI, nº de teléfono, etc.). Estos datos son los que llamamos atributos, y tienen un valor asociado ("Pepe", "C/Frasquito N/16", "333333354", "45555556", etc....).

Cualquier entidad debe de cumplir estas características:

- > Todas las ocurrencias deben de tener el mismo número de atributos, sin importar que carezcan de algún valor. Más adelante veremos como trata esta situación.
- > No puede haber dos ocurrencias que tengan los mismos valores para todos sus atributos, porque sería una duplicidad.

Cuando queremos representar gráficamente una entidad, se hace del siguiente modo: se dibuja un rectángulo con el nombre la entidad en su interior (en mayúsculas siempre que sea posible) y entonces tenemos dos modos de representar los atributos de dicha entidad, mediante óvalos o mediante círculos pequeños. Si realizamos los atributos como óvalos, el nombre de cada atributo va dentro, mientras que, si son círculos, van al lado. Un ejemplo sería:

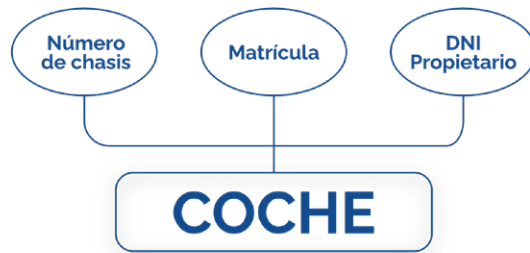


Imagen 2. Representaciones gráficas de entidades y atributos

Podemos diferenciar dos tipos de entidades:

- > **Fuerte o regular.** Es totalmente independiente de las otras. Son la mayoría de las entidades.
- > **Débil.** Sus ocurrencias dependen de las existencias de las ocurrencias de otra entidad distinta.

Las entidades débiles deben de serlo con respecto a todas las entidades con las que se relacionan, y su representación es la siguiente:



Imagen 3. Representación gráfica de una entidad débil.

## 2.2.2. Relación

Las ocurrencias de varias entidades distintas pueden tener aspectos en común, lo que da lugar a una relación entre ellas. Las relaciones se nombran usando la tercera persona de singular de indicativo de un verbo y un ejemplo podría ser entre "camionero" y "camión" con la relación "conduce".

La representación gráfica de una relación se realiza dibujando un rombo con el nombre de la relación dentro, como el ejemplo siguiente:



Imagen 4. Representación gráfica de una relación.

Dependiendo del número de entidades en las que sus concurrencias tienen alguna relación, tenemos varios tipos de relaciones:

- > **Binarias.** Relacionan entre sí ocurrencias de solamente dos entidades distintas. Por ejemplo, podríamos hablar de un alumno que cursa alguna materia:



Imagen 5. Relación binaria

- > **Ternarias.** Se relacionan las ocurrencias de tres entidades distintas entre sí.

Por ejemplo, siguiendo con el ejemplo anterior, un alumno cursa una materia que pertenecerá a un ciclo específico:

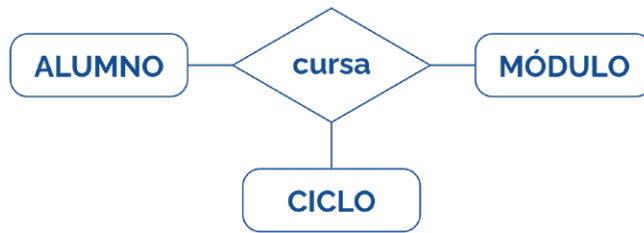


Imagen 6. Relación ternaria

- > **N-arias.** Se puede dar el caso dependiendo de la complejidad de nuestra base de datos de que tengamos que establecer una relación de más de tres entidades al mismo tiempo. Podemos aplicar como ejemplo el mismo anterior, en el que alumno cursa el módulo de un ciclo en específico y en un centro en particular:

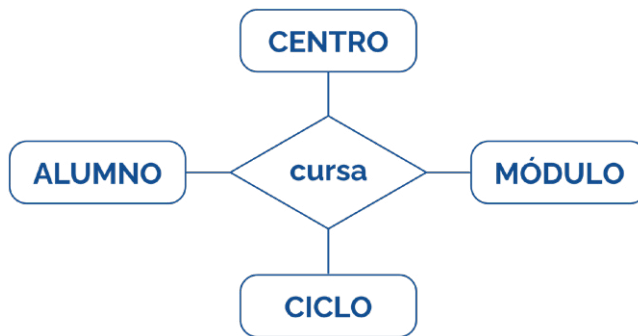


Imagen 7. Relación cuaternaria

Un pequeño consejo para el diseño de las bases de datos es que identificaremos relaciones ternarias o n-arias cuando la acción que se realice afecte de manera simultánea a todas las ocurrencias de las distintas entidades implicadas.

- > **Reflexivas.** Se relacionan entre sí ocurrencias que tienen cabida en la misma entidad. Esto sucede porque, aunque estén dentro de una misma entidad, la realidad es que estas ocurrencias ocupan funciones distintas. Tenemos el ejemplo en el que hay varios alumnos en una clase y uno debe ser el delegado de todos los demás. No hay necesidad de crear otra entidad porque lo único que cambia es si es delegado o no, pero todo lo demás se conserva igual y por lo tanto se establece una relación reflexiva:

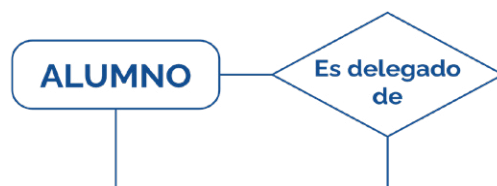


Imagen 8. Relación reflexiva.



### 2.2.3. Cardinalidad y modalidad

Si lo que queremos es dotar de un cierto contenido lógico y estructurado a nuestra relación, tenemos que especificar como se relacionan las ocurrencias de las entidades en cuestión. Esto se realiza gracias a que hay una serie de ámbitos, límites y restricciones que están preestablecidos.

La **cardinalidad o tipo de correspondencia** nos va a indicar el número de máximo de datos de una ocurrencia en relación con la otra ocurrencia con la que se relacione. La cardinalidad maneja varias situaciones para poder abarcar el máximo posible:

- > **1:N (uno a ene/uno a muchos).** En este caso la primera ocurrencia en la relación, la del lado donde apunta el 1, se puede relacionar con muchos del extremo, mientras que la ocurrencia del lado N, solo puede relacionarse con una como máximo del otro extremo.

Vamos a imaginar que tenemos una asignatura que puede ser impartida por varios profesores, pero un profesor solo puede impartir de manera normal una asignatura, pues su representación gráfica sería la siguiente:



Imagen 9. Relación 1: N.

- > **M:N (eme a ene/muchos a muchos).** En ambos casos, ambas ocurrencias pueden relacionarse al mismo tiempo con varias ocurrencias del extremo opuesto.

Tenemos la casuística de que un alumno puede tener varios módulos que cursar, pero un mismo módulo puede ser cursado por varios alumnos, se representaría gráficamente del siguiente modo:



Imagen 10. Relación M: N.

- > **1 : 1 (uno a uno).** Una ocurrencia de un extremo solo se puede relacionar con otra del extremo opuesto. Lo mismo en ambas direcciones. En este caso sabemos que un director puede dirigir un solo departamento, y un departamento solo puede tener un director, como el siguiente ejemplo:



Imagen 11. Relación 1:1.

La cardinalidad nos ayuda a establecer los límites superiores en una relación, pero realmente no existe una obligatoriedad de que sea justo esa la cantidad necesaria.

Para solucionar este problema tenemos la *modalidad*. Esta característica nos va a permitir establecer dentro de la cardinalidad unos límites de la propia ocurrencia, es decir, unos máximos y unos mínimos. Esta modalidad la indicamos a ambos lados de la relación, es decir, en cada ocurrencia y el valor máximo siempre va a coincidir con el valor que se haya asignado mediante la cardinalidad de la relación. Hay varios tipos, son los siguientes:

- > **(0,1) (cero a uno)**. La ocurrencia de la entidad contraria se puede relacionar con ninguna ocurrencia de esta entidad y como máximo con una, nunca con varias.
- > **(1,1) (uno a uno)**. Sí o sí, la ocurrencia contraria en la relación debe de relacionarse con una ocurrencia de nuestra entidad. Solo con una.
- > **(1,N) (uno a ene)**. La ocurrencia contraria se va a relacionar con al menos una ocurrencia de nuestra entidad y puede que con varias.
- > **(0,N) (cero a ene)**. En la entidad contraria, la ocurrencia se relacionará con nuestra entidad pudiendo relacionarse con ninguna o con varias ocurrencias a la vez, es indiferente.

Un ejemplo de relación binaria 1:N donde tenemos en un lado (0,1) y en otro (0,N) correspondiente a una tienda de ordenadores:



Imagen 12. Relación (0,1) a (0,N).

En este ejemplo, su modalidad se lee en el lado opuesto, es decir, se explica del siguiente modo:

- > Un cliente puede comprarse ninguno o varios ordenadores, pero no tiene por qué comprar ninguno y puede que se compre más de uno.
- > Un ordenador (uno en específico, no un modelo), puede ser comprado por un cliente únicamente, pero también puede ser que no sea comprado nunca, por eso lo de este tipo de modalidad.

Este otro ejemplo refleja una situación distinta:



Imagen 13. Relación (1,1) a (1,N).

En el ejemplo anterior, tenemos lo siguiente:

- > Un autor puede escribir uno o varios libros (mínimo uno, si no, no sería autor).
- > Mientras, un libro únicamente puede ser escrito por un autor, y siempre será mínimo uno y máximo uno porque si no se escribe, no existe.

Un ejemplo donde tenemos una relación M:N y en un extremo su modalidad que es (0,N) y (1,N). El siguiente es cuando un alumno cursa un módulo. Este alumno puede cursar un módulo mínimo (si no, no sería alumno), pero también puede cursar varios al mismo tiempo.

También tenemos que saber que un módulo puede ser cursado por varios alumnos, pero también puede ser que no lo curse ninguno porque nadie se haya matriculado de ese módulo, por mucho que exista:

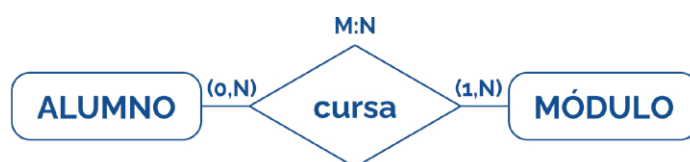


Imagen 14. Relación (0,N) a (1,N).

Un último ejemplo en una relación que acapara cuatro entidades y en las que la empezando de arriba abajo tienen la siguiente modalidad: (1,N) por que un alumno puede cursar el ciclo en uno o varios centros (varios ciclos a la vez), (1,N) como hemos dicho antes se pueden cursar uno o varios módulos, (1,N) se puede cursar más de un ciclo al mismo tiempo pero mínimo uno y (0,N) porque como antes, puede que no haya alumnos y puede que haya varios que cursen distintos módulos y ciclos.

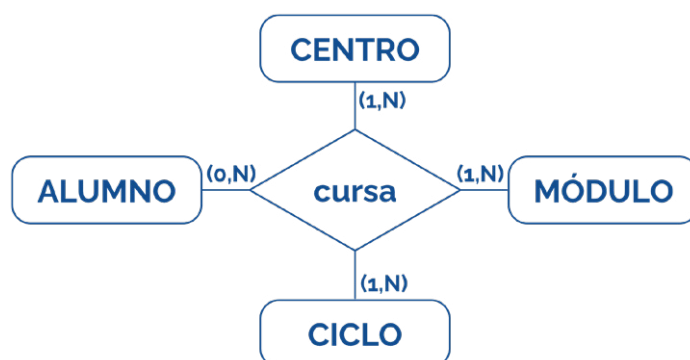


Imagen 15. Relación cuaternaria.

## 2.2.4. Atributos de relación

Hay ocasiones en las que ciertas propiedades o alguna característica afecta al mismo tiempo a dos entidades y a su vez tiene que ver con la relación que hay formada entre dichas entidades. Es en este momento cuando se crea un atributo de relación y afecta directamente a esta, lo que hace que afecte al mismo tiempo a las dos o más entidades que toman parte de la relación. Se representa como un atributo normal pero no en cada entidad, sino en la relación. En este ejemplo tenemos que para identificar a un empleado tendremos en cuenta su código de empleado, su NIF, su nombre y su salario. En cuanto a curso, se almacena el código del curso, el nombre y el número de horas que dura.

Ahora bien, tenemos que almacenar también cuando empezó dicho curso, que es tanto cuando lo empezó el empleado y cuando comienza como tal. Es por eso por lo que este atributo se almacena en la relación y no en cada entidad, porque es común.

Gráficamente se representa igual que en una entidad, pero pertenece a la relación:

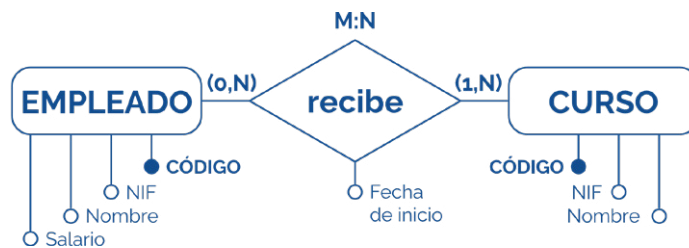


Imagen 16. Atributo de relación.

## 2.2.5. Clave primaria y claves candidatas

Siempre que tengamos una entidad, habrá un atributo o un conjunto de atributos que con su valor nos ayudarán a identificar la entidad de todas las demás.

En cada entidad habrá un atributo que sea único para cada dato que se introduzca y que nos ayude a diferenciarlo de todos los demás, por ejemplo, el DNI de cada persona es único para ella y no puede haber dos iguales. Esto es lo que llamamos la clave primaria. Nos ayuda a identificar un dato de la entidad como antes hemos dicho y es única, no puede haber dos iguales.

En el siguiente ejemplo tenemos un socio de alguna empresa u organización que tiene los siguientes atributos: código de socio, nombre, tipo, domicilio, teléfono y apellido.

Bien, si lo que queremos es distinguirlo de los demás, veremos que solo hay una cosa que será única para él y no para ningún otro socio, que es el código de socio.

Si representamos mediante círculos los atributos, la clave primaria se representa como un círculo relleno, mientras que, si lo representamos mediante óvalos, esta clave es la palabra que esté subrayada. Aquí vemos una descripción gráfica:



Imagen 17. Representaciones gráficas de clave primaria.

No siempre es posible identificar los datos por un solo atributo porque este se puede repetir. Pero puede que no haya ninguna que excluya a las demás. En este caso de manera casi general, es una combinación de varios atributos los que hacen que haya una identificación única. Ahora tendríamos una clave primaria formada por varios atributos. El ejemplo de a continuación nos muestra la entidad 'empleado' en la que tenemos como clave primaria tanto código de usuario como código de empleado. Esto se debe a que en solitario no son únicas, pero su combinación sí, porque una va ligada a la otra.



Imagen 18. Clave primaria compuesta.



### 2.2.6. Ejemplo de diagrama entidad/relación

Tenemos un sistema de venta, donde necesitamos que se tengan en cuenta los siguientes requisitos:

- > Tenemos unos **proveedores** que nos **ofrecen** distintos **productos**. De estos proveedores tenemos que almacenar su **código de proveedor**, **nombre**, **web**, **teléfono** y **dirección**.
- > De cada **producto** almacenamos su **id de producto**, su **nombre**, el **stock** que hay y su **precio**.
- > Los **productos** pertenecen a una cierta **categoría**. De la que almacenamos el **id de la categoría**, su **nombre** y una **breve descripción**.
- > Además, los **productos** participan en una **venta**, y tenemos que almacenar **coste total del producto** venido, el **precio por unidad** y la **cantidad de venta** de cada producto.
- > De la **venta** almacenaremos el **número de factura**, el **descuento** que se realice, la **fecha** y el **coste final**.
- > Estas **ventas** son porque ha **comprado** un **cliente**, del que almacenaremos el **código del cliente**, el **nombre**, su **dirección** y su **teléfono**.

Esto quedaría finalmente del siguiente modo:

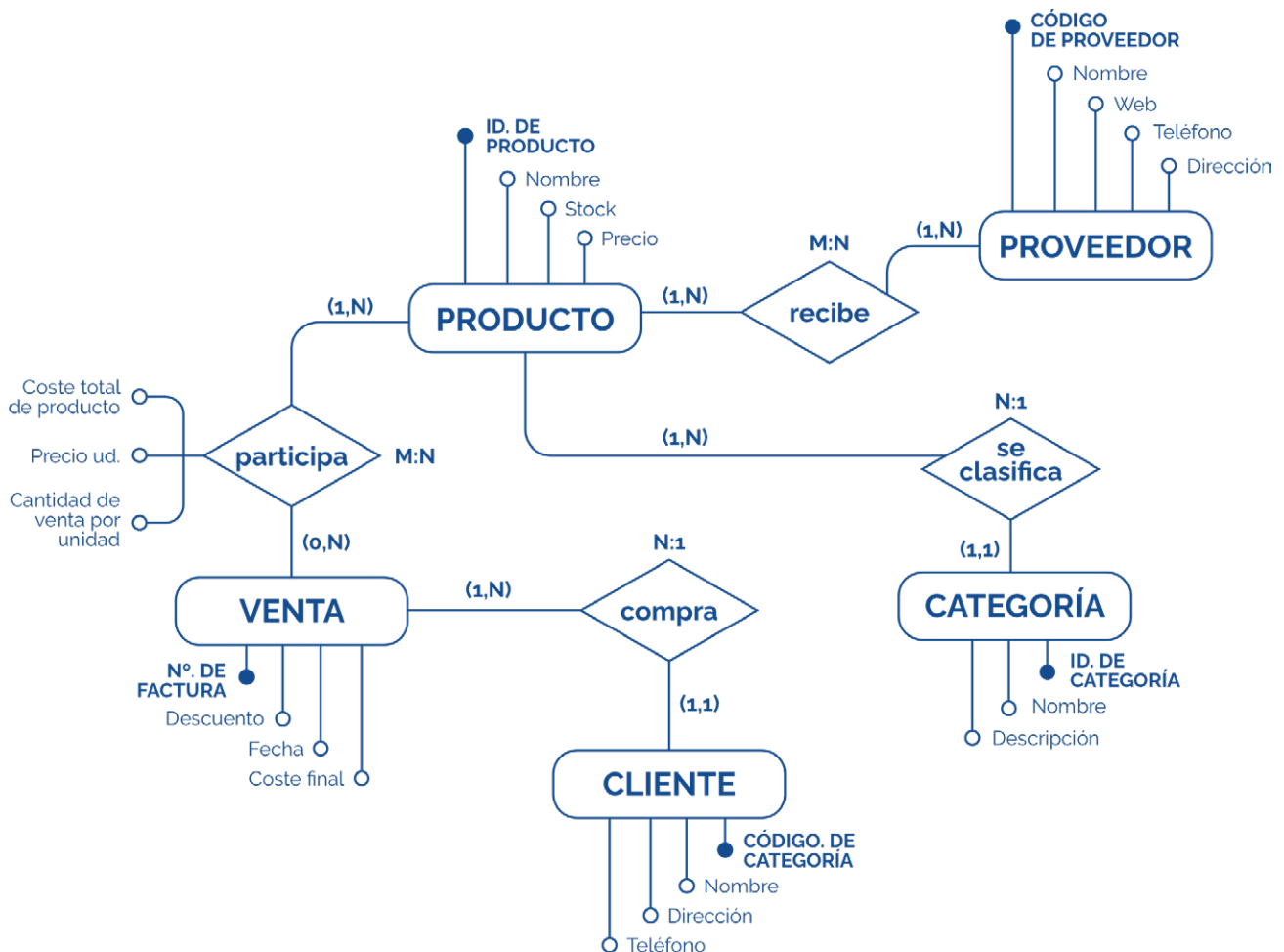


Imagen 19. Ejemplo de diagrama entidad/relación.



Vamos a explicar lo que hemos realizado:

- > Se han creado las entidades 'PROVEEDOR', 'PRODUCTO', 'CATEGORIA', 'VENTA' y 'CLIENTE'.
- > Se añaden las relaciones entre ellos, como podemos ver hay entre proveedor y producto, entre producto y categoría, entre producto y venta y entre venta y cliente.
- > Se añaden los atributos de las entidades como hemos dicho que queremos almacenar.
- > Se seleccionan las claves primarias de dichas entidades.
- > Añadimos, porque en este caso los hay, los atributos de relación.

Si explicamos en más profundidad, tenemos lo siguiente:

- > 'PROVEEDOR' tiene como clave primaria código de proveedor, porque es lo que lo identifica.
- > 'PRODUCTO' tiene como clave primaria id de producto.
- > Se establece entre los dos la relación 'tiene' que es muchos a muchos. Un producto puede ser traído por uno o varios proveedores, pero siempre por uno mínimo. A la misma vez, un proveedor nos puede traer uno o más productos de igual modo.
- > Ahora añadimos la entidad 'CATEGORIA' donde su clave primaria será id de categoría.
- > Entre 'PRODUCTO' y 'CATEGORIA' se establece la relación 'se clasifica' que es muchos a uno. Una categoría puede englobar uno o varios productos, pero mínimo uno, de manera lógica. De modo similar, un producto solo puede pertenecer a una categoría, sí o sí a una.
- > Siguiendo con las relaciones de producto, se crea la entidad 'VENTA', que tiene como clave primaria número de factura.
- > Entre 'PRODUCTO' y 'VENTA' creamos la relación 'participa'. En esta relación guardaremos los atributos de relación: coste total de producto, precio unidad y cantidad de venta por unidad.
- > La relación es muchos a muchos ya que un producto puede venderse en ninguna o varias ventas, y una venta mínima venderá un producto, pero puede vender varios.
- > Por último, creamos la entidad 'CLIENTE' cuya clave primaria será código de cliente.
- > Entre 'CLIENTE' y 'VENTA' se establecerá la relación 'compra', que es muchos a uno porque una venta será por un único cliente, pero un mismo cliente puede realizar varias ventas, mínimo una.



# 2.3.

## Diagrama entidad/relación extendido

Del modo que se planteó en un primer momento el diagrama entidad/relación había una serie de cuestiones a las que no daba solución con respecto a los problemas que el modelo de conceptual de datos plantea. Esto hace que algunos expertos añadiesen aún más restricciones para que su magnitud semántica se vea encarecida en ciertos casos. Realmente esto no es un hecho que se lleve a cabo de manera unánime en el diseño de las bases de datos, pero tenemos algunos elementos muy reconocibles del diagrama de entidad/relación extendido.

- > **Exclusividad:** si existe una relación, no puede existir otra. Un ejemplo sería el siguiente, en el que tenemos que un vehículo puede o consumir litros de gasoil o gastar litros de gasolina, pero nunca las dos al mismo tiempo:

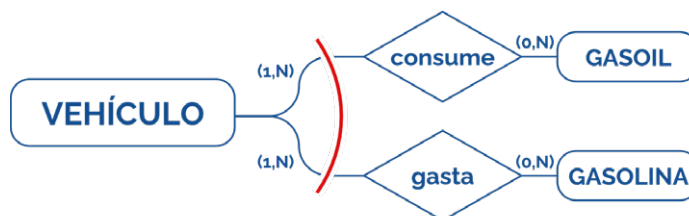


Imagen 20. Exclusividad.

- > **Jerarquías:** Existe una primera entidad con una ocurrencia y se relaciona con varias ocurrencias de distintas entidades al mismo tiempo. Esta entidad se llama supertipo y tienen atributos compartidos con las demás entidades o subtipos, mientras que las entidades de menor jerarquía cuentan con estos atributos del supertipo y con otros propios. Por ejemplo, tenemos una empresa con empleados, los cuales pueden ser o asalariados fijos o estar contratados por horas. Estos empleados tienen algunas cosas en común, pero otras que no, y en este caso se usaría una jerarquía como la siguiente mostrada:

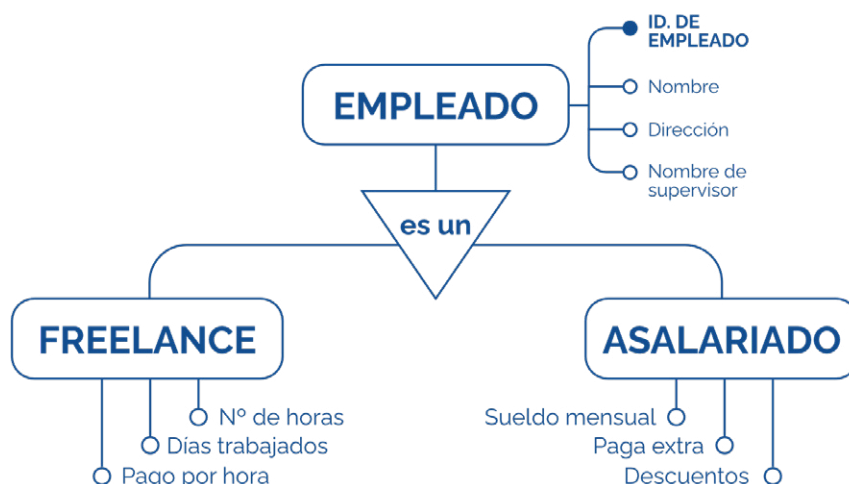


Imagen 21. Jerarquía.



Casi siempre que hablamos de jerarquía partiremos de la entidad supertipo y por lo general las entidades subtipo nos ayudarán a crear una *especialización* que ayudan a dar un sentido más específico a la información que se agregue. Por otra parte, si el inicio de la jerarquía reside en los subtipos se hablará de una *generalización* que nos ayudará a crear un conjunto de entidades para su mejor entendimiento.

- > **Agregación.** En este caso varias ocurrencias de entidades distintas en conjunto crean una ocurrencia para otra entidad distinta. Un ejemplo sería este en el que agruparíamos las entidades empresa y solicitante en otra que se puede llamar entrevista:

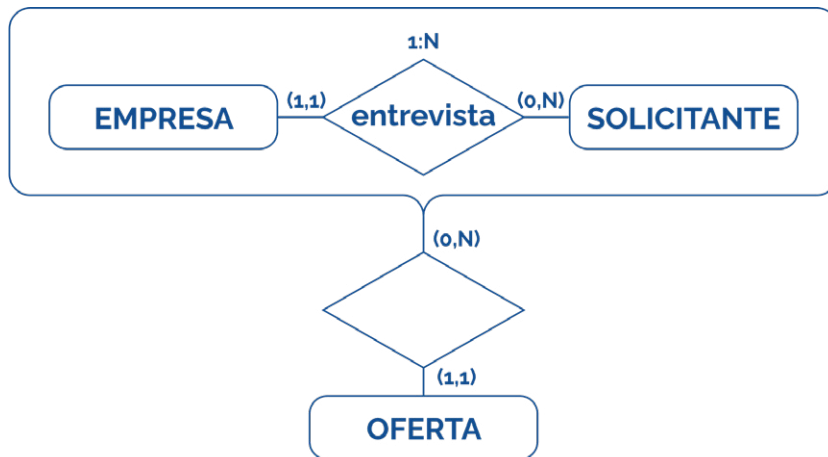


Imagen 22. Agregación.





 [www.universae.com](http://www.universae.com)

