

## APUNTES

### BASES DE DATOS: UNIDAD 2

El proceso de conversión del diseño conceptual de un Diagrama ER al **modelo relacional** implica varias fases clave:

- **Conversión de entidades en tablas:** Cada entidad identificada en el Diagrama ER se convierte en una tabla dentro del modelo relacional. Los atributos de la entidad se transforman en columnas de la tabla correspondiente, y cada fila de la tabla representa una instancia o registro de esa entidad.
  - **Definición de claves primarias:** Durante la conversión, es fundamental determinar cuáles son los atributos que funcionarán como **clave primaria** en cada tabla. La clave primaria es un identificador único para cada registro en la tabla, y su correcta definición garantiza la unicidad y facilita la búsqueda eficiente de datos.
  - **Conversión de relaciones:** Las relaciones entre entidades, tal como se representan en el Diagrama ER, deben ser mapeadas en el modelo relacional. Las relaciones **uno a uno (1:1)**, **uno a muchos (1:N)** y **muchos a muchos (M:N)** se traducen mediante el uso de **claves externas** y tablas adicionales, en el caso de relaciones M:N, para garantizar la correcta representación de las asociaciones entre entidades.
- 

El proceso de transformación de relaciones desde un modelo Entidad-Relación (ER) al modelo relacional requiere una serie de pasos específicos y precisos. Esto es esencial para asegurar que las relaciones entre las entidades se representen adecuadamente en el modelo relacional y que se mantenga la **integridad referencial**. A continuación, se explican las reglas clave para la transformación de los diferentes tipos de relaciones, como las **muchos a muchos (M:N)**, **uno a muchos (1:N)** y **uno a uno (1:1)**.

#### 1. Relaciones Muchos a Muchos (M:N)

Las **relaciones muchos a muchos (M:N)** en un diagrama ER representan una situación en la que múltiples instancias de una entidad están asociadas con múltiples instancias de otra entidad. Sin embargo, en el modelo relacional, las relaciones M:N no pueden ser representadas directamente entre dos tablas debido a las restricciones de diseño relacional. Para resolver este problema, es necesario crear una **tabla intermedia** (a menudo denominada tabla de unión o tabla de relación) que actúa como un conector entre las dos entidades involucradas.

- **Tabla intermedia:** Esta tabla intermedia contendrá las **claves primarias** de ambas entidades que participan en la relación, y esas claves se utilizarán como **claves externas** en la tabla intermedia. La tabla intermedia, por tanto, crea una relación entre las dos entidades principales, manteniendo la **integridad referencial**. Además de las claves externas, si la relación M:N tiene atributos propios (por ejemplo, fecha de inicio o tipo de asociación), estos atributos se transforman en **columnas** de la tabla intermedia.
- **Clave compuesta:** En la mayoría de los casos, la clave primaria de la tabla intermedia será una **clave compuesta**, formada por la combinación de las claves primarias de ambas

entidades que participan en la relación. Esto asegura que no haya duplicados en la tabla intermedia, es decir, que cada combinación única de las dos entidades involucradas solo aparezca una vez.

- **Atributos de la relación:** Si la relación entre las entidades tiene atributos adicionales (por ejemplo, en una relación entre "Alumno" y "Curso", podría haber un atributo como "Fecha de inscripción"), estos atributos se añaden como columnas en la tabla intermedia.

**Ejemplo:** Supongamos que en un sistema de gestión de universidades, existe una relación M:N entre las entidades "Estudiante" y "Curso", donde un estudiante puede inscribirse en múltiples cursos, y un curso puede tener múltiples estudiantes inscritos. En el modelo relacional, se creará una tabla intermedia llamada "Inscripción" que contendrá:

- Clave primaria del "Estudiante" (como clave externa).
- Clave primaria del "Curso" (como clave externa).
- Atributos adicionales, como "Fecha de inscripción" o "Nota final".

La clave compuesta en la tabla "Inscripción" estará formada por la combinación de "ID\_Estudiante" e "ID\_Curso", lo que garantiza que un estudiante no pueda inscribirse más de una vez en el mismo curso sin duplicar datos.

## 2. Relaciones Uno a Muchos (1:N)

Las **relaciones uno a muchos (1:N)** son extremadamente comunes en los sistemas de bases de datos y representan una situación en la que una instancia de una entidad está relacionada con múltiples instancias de otra entidad. El proceso de transformación de estas relaciones en el modelo relacional es más sencillo que el de las relaciones M:N, ya que no requiere una tabla intermedia en la mayoría de los casos.

Existen dos enfoques principales para manejar relaciones 1:N:

- **Propagación de la clave:** El método más común y eficiente para manejar las relaciones uno a muchos es **propagar la clave primaria** de la entidad del lado "uno" como una **clave externa** en la tabla que representa la entidad del lado "muchos". Este enfoque asegura que cada registro en la tabla del lado "muchos" esté relacionado con un único registro de la tabla del lado "uno", garantizando la integridad referencial.
  - **Clave externa:** La clave primaria de la tabla que representa la entidad "uno" se convierte en una clave externa en la tabla del lado "muchos". Esto crea una relación explícita entre ambas tablas.
  - **Integridad referencial:** Se asegura que cada registro del lado "muchos" tenga una referencia válida en el lado "uno", evitando inconsistencias en los datos.

**Ejemplo:** En un sistema de gestión de empresas, una relación 1:N podría existir entre las entidades "Departamento" y "Empleado", donde un departamento puede tener muchos empleados, pero cada empleado pertenece a un solo departamento. En el modelo relacional, la tabla "Empleado" incluiría una columna llamada "ID\_Departamento", que actúa como clave externa referenciando la clave primaria de la tabla "Departamento". Esto asegura que cada empleado esté asociado con un único departamento.

- **Tabla intermedia opcional:** Aunque es menos común, en algunas situaciones se puede optar por crear una **tabla intermedia** para gestionar una relación 1:N. Esto suele hacerse si existe la posibilidad de que la relación evolucione hacia una relación M:N en el futuro, o si se necesitan atributos adicionales para describir la relación.

**Ejemplo:** En un sistema de ventas, podría haber una relación entre "Cliente" y "Pedido". Aunque actualmente es 1:N (un cliente puede hacer muchos pedidos), la empresa podría decidir que la relación cambie a M:N en el futuro (por ejemplo, si varios clientes pueden hacer un pedido conjunto), y se puede optar por crear una tabla intermedia como "Cliente\_Pedido" desde el principio.

### 3. Relaciones Uno a Uno (1:1)

Las **relaciones uno a uno (1:1)** representan una situación en la que cada instancia de una entidad está asociada con una única instancia de otra entidad, y viceversa. Estas relaciones son relativamente raras, pero cuando existen, su transformación en el modelo relacional puede realizarse de varias maneras, dependiendo de las restricciones de integridad y de los requisitos de diseño.

Existen dos métodos principales para manejar relaciones 1:1 en el modelo relacional:

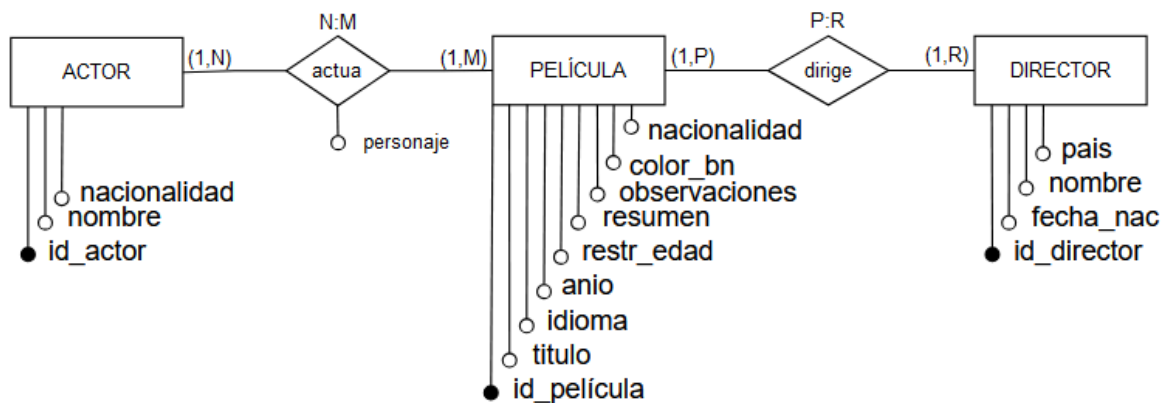
- **Propagación de clave primaria:** En este enfoque, la **clave primaria** de una de las entidades se convierte tanto en la clave primaria como en la **clave externa** de la otra entidad. Esto garantiza que haya una correspondencia directa y única entre ambas tablas. Este método es eficiente y asegura que las dos tablas estén estrechamente vinculadas.

**Ejemplo:** En un sistema médico, podríamos tener una relación 1:1 entre las entidades "Paciente" y "Historia Clínica", donde cada paciente tiene una única historia clínica, y cada historia clínica pertenece a un solo paciente. En este caso, el "ID\_Paciente" podría ser tanto la clave primaria como la clave externa en la tabla "Historia Clínica", asegurando que cada historia clínica esté asociada de manera única con un paciente.

- **Tablas separadas con claves externas:** En algunos casos, es deseable mantener ambas tablas separadas por razones de diseño (rendimiento, seguridad, etc.). En este enfoque, ambas entidades mantienen sus propias claves primarias y se utiliza una **clave externa** en cada tabla para referenciarse mutuamente. Este método es más común cuando las entidades deben mantenerse separadas por razones específicas, como restricciones de privacidad o rendimiento.

**Ejemplo:** En una base de datos gubernamental, podríamos tener una relación 1:1 entre "Ciudadano" y "Registro de Votante". Para asegurar una mayor seguridad y privacidad, podría ser preferible mantener ambas tablas separadas, con "ID\_Ciudadano" como clave externa en la tabla "Registro de Votante" y viceversa.

## EJEMPLO RELACIÓN M:N



- **DIRECTOR**(ID, FECHA\_NACIMIENTO, PAIS\_ORIGEN)
  - **PELICULA**(ID, TITULO, AÑO, NACION, IDIOMA, COLOR, RESTRICCIÓN, RESUMEN, OBSERVACIÓN)
  - **ACTOR**(ID, NOMBRE, NACIONALIDAD)
- 
- ❖ **DIRECTOR\_DIRIGE\_PELICULA**(ID DIRECTOR, ID PELICULA)
  - ❖ **ACTOR\_PARTICIPA\_PELICULA**(ID ACTOR, ID PELICULA, PERSONAJE)

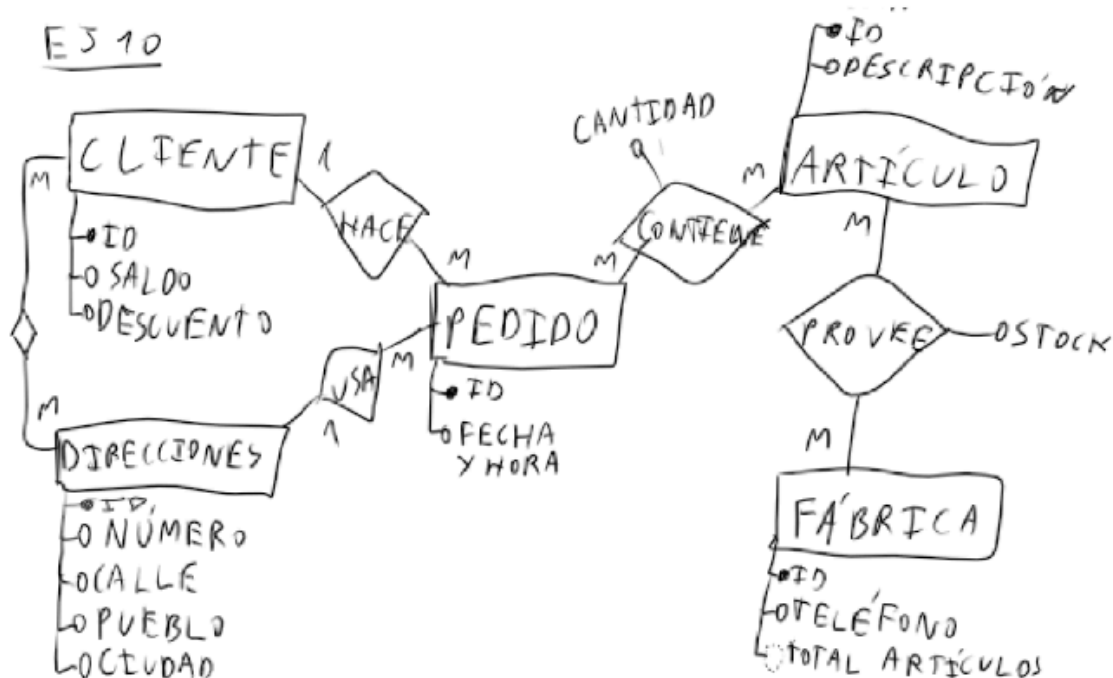
En este ejemplo, las dos relaciones existentes tienen una cardinalidad de M:N. Por ello, ambas relaciones se convierten en tablas intermedias con claves primarias compuestas.

Así mismo, **se añaden los atributos de la relación a la tabla pertinente**. En este caso, PERSONAJE es un atributo de la relación PARTICIPA entre ACTOR y PELÍCULA.

**Las claves primarias deben estar subrayadas y se le debe asignar un color a cada una** para mayor claridad. No hay representación de atributos candidatos, compuestos, derivados o multivaluados.

## EJEMPLO RELACIÓN 1:N, M:N, VARIAS RELACIONES A UNA ENTIDAD

### EJERCICIO 10

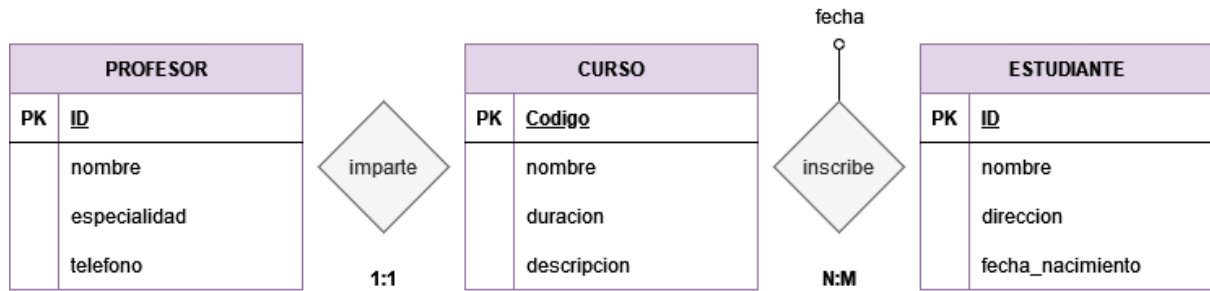


- **CLIENTE**(ID, SALDO, DESCUENTO)
  - **DIRECCION**(ID, NUMERO, CALLE, PUEBLO CIUDAD)
  - **PEDIDO**(ID, FECHA\_HORA, ID\_CLIENTE, ID\_DIRECCION)
  - **ARTICULO**(ID, DESCRIPCION)
  - **FABRICA**(ID, TELEFONO, TOTAL\_ARTICULOS)
- 
- ❖ **CLIENTE\_TIENE\_DIRECCIONES**(ID\_CLIENTE, ID\_DIRECCION)
  - ❖ **PEDIDO\_CONTIENE\_ARTICULO**(ID\_PEDIDO, ID\_ARTICULO, CANTIDAD)
  - ❖ **ARTICULO\_PROVEE\_FABRICA**(ID\_ARTICULO, ID\_FABRICA, STOCK)

En este ejemplo existen dos relaciones de 1:N. La primera, CLIENTE HACE PEDIDO. La segunda, DIRECCIÓN USA PEDIDO. En ambos casos, PEDIDO es la entidad que representa una relación de M (muchos). CLIENTE y DIRECCIÓN representan “el lado 1” respectivamente. Por lo tanto, sus claves primarias se propagan a la entidad/tabla PEDIDO.

**Las claves externas mantienen el color asignado pero no se subrayan, ya que no son atributos claves de la entidad/tabla PEDIDO.**

## EJEMPLO RELACIÓN 1:1



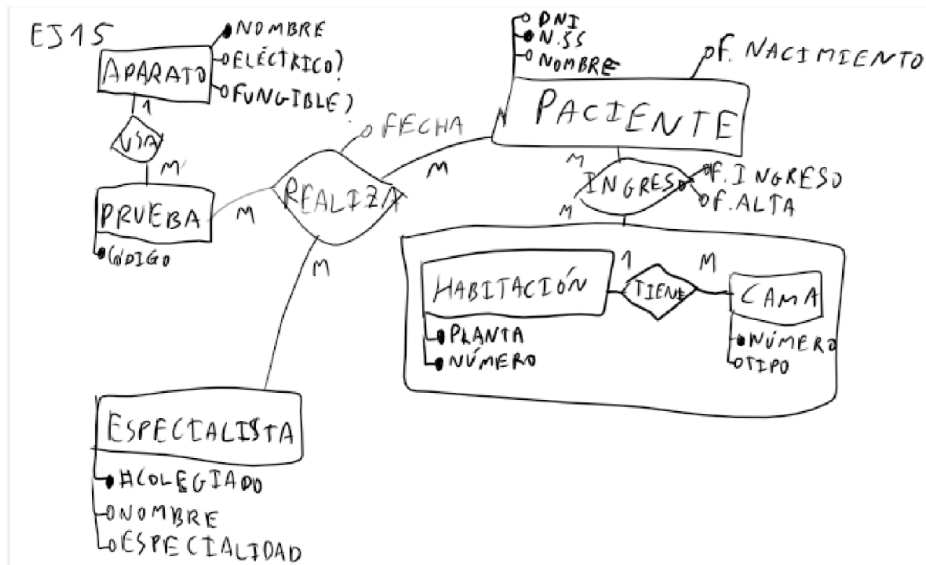
- **PROFESOR**(ID, NOMBRE, ESPECIALIDAD, TELEFONO)
  - **CURSO**(CODIGO, NOMBRE, DURACION, DESCRIPCION, ID\_PROFESOR)
  - **ESTUDIANTE**(ID, NOMBRE, DIRECCION, FECHA\_NACIMIENTO)
- ❖ **ESTUDIANTE\_INSCRIBE\_CURSO**(ID\_ESTUDIANTE, ID\_CURSO, FECHA)

---

Lo común es que en las relaciones de 1:1, una de las claves primarias sea simultáneamente clave primaria y clave externa. **Es indiferente cuál de ellas se propague.**

## EJEMPLO CON RELACIÓN TERNARIA

### EJERCICIO 15



- **PACIENTE**(NUMERO\_SS, DNI, NOMBRE, FECHA\_NACIMIENTO)
  - **HABITACION**(PLANTA, NUMERO)
  - **CAMA**(NUMERO, TIPO, PLANTA, NUMERO\_HABITACION)
  - **APARATO**(NOMBRE, ELECTRICO, FUNGIBLE)
  - **PRUEBA**(CODIGO, NOMBRE, APARATO)
  - **ESPECIALISTA**(NUMERO\_COLEGIADO, NOMBRE, ESPECIALIDAD)
- 
- ❖ **PACIENTE\_INGRESA\_HABITACION\_CAMA**(NUMERO\_SS, NUMERO\_CAMA, FECHA\_INGRESO, FECHA\_ALTA)
  - ❖ **ESPECIALISTA\_REALIZA\_PRUEBA\_PACIENTE**(NUMERO\_COLEGIADO\_ESPECIALISTA, CODIGO\_PRUEBA, NUMERO\_SS\_PACIENTE, FECHA)

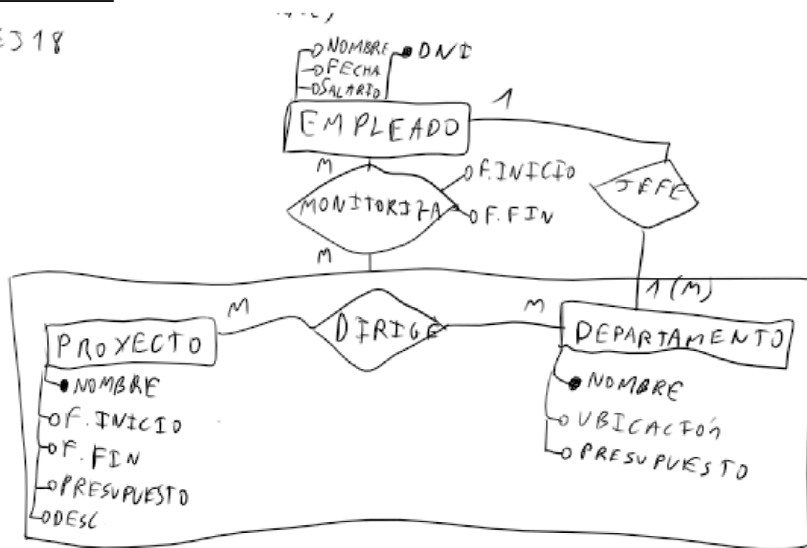
La relación ternaria se convierte en una tabla intermedia que contiene las tres entidades con su relación.

HABITACION TIENE CAMA no se representaría como una agregación, ya que es una relación de 1:N. Por lo tanto, en este caso particular, no es necesaria una tabla intermedia. La parte M de esta relación, es decir, CAMA es la información que nos hace falta para la tabla PACIENTE\_INGRESA\_HABITACION\_CAMA. Hacer referencia a HABITACION no es necesario.

## EJEMPLO DE RELACIÓN COMO ATRIBUTO

### EJERCICIO 18

EJ 18



- **EMPLEADO**(DNI, NOMBRE, FECHA, SALARIO)
- **PROYECTO**(NOMBRE, FECHA\_INICIO, FECHA\_FIN, PRESUPUESTO, DESCRIPCION)
- **DEPARTAMENTO**(NOMBRE, UBICACION, PRESUPUESTO, DNI\_JEFE)
- ❖ **EMPLEADO\_MONITORIZA\_PROYECTO\_DEPARTAMENTO**(DNI\_EMPLEADO, NOMBRE\_PROYECTO, NOMBRE\_DEPARTAMENTO, FECHA\_INICIO, FECHA\_FIN)
- ❖ **DEPARTAMENTO\_DIRIGE\_PROYECTO**(NOMBRE\_DEPARTAMENTO, NOMBRE\_PROYECTO)

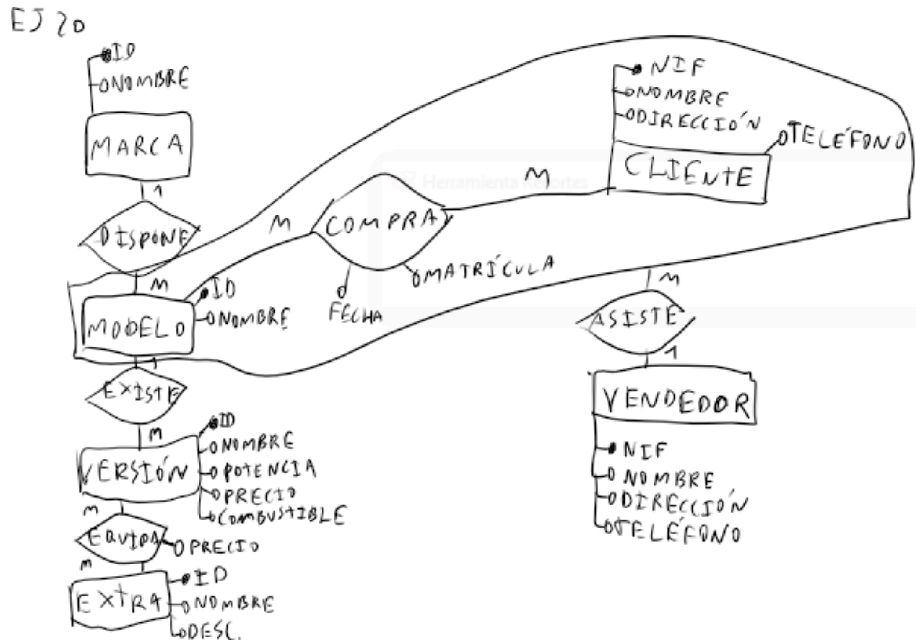
---

La relación EMPLEADO JEFE DEPARTAMENTO, no se convierte en una tabla más. Si no que la clave primaria de EMPLEADO, DNI, se propaga con el mismo color pero añadiendo JEFE (DNI\_JEFE). ID\_JEFE puede estar tanto en la tabla EMPLEADO como en la de DEPARTAMENTO.



## EJEMPLO CON AGREGACIÓN M:N

### EJERCICIO 20

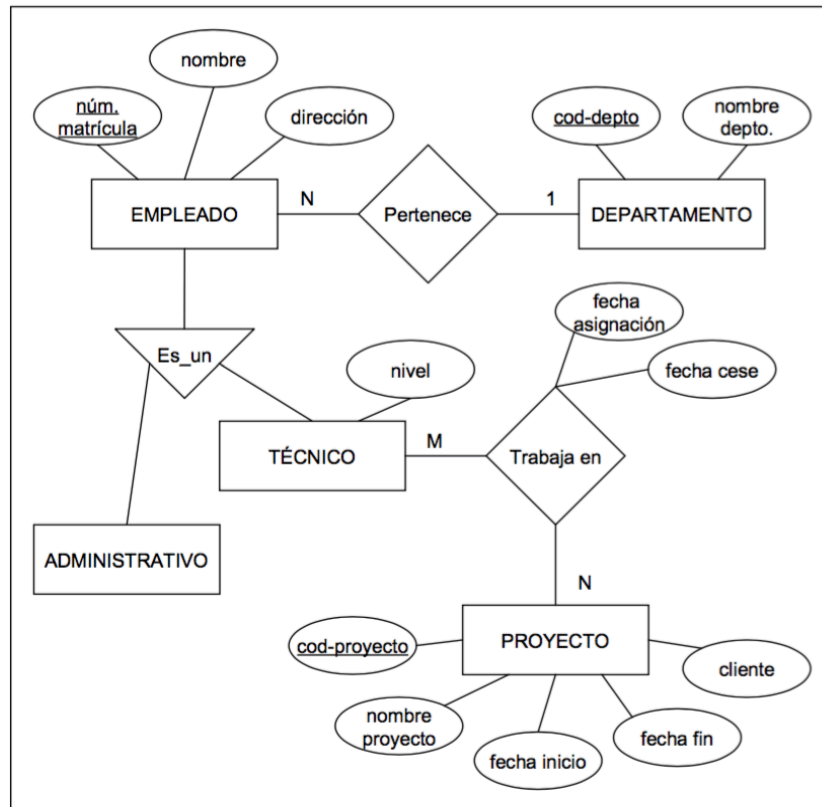


- **MARCA**(ID, NOMBRE)
  - **MODELO**(ID, NOMBRE, ID\_MARCA)
  - **CLIENTE**(NIF, NOMBRE, DIRECCION, TELEFONO)
  - **VERSION**(ID, NOMBRE, POTENCIA, PRECIO, COMBUSTIBLE, ID\_MODELO)
  - **EXTRA**(ID, NOMBRE, DESCRIPCION)
  - **VENDEDOR**(NIF, NOMBRE, DIRECCION, TELEFONO)
- 
- ❖ **CLIENTE\_COMPRA\_MODELO**(ID\_MODELO, NIF\_CLIENTE, FECHA, MATRICULA, NIF\_VENDEDOR)
  - ❖ **VERSION\_EQUIPA\_EXTRA**(ID\_VERSION, ID\_EXTRA, PRECIO)

---

La agregación presente (M:N), CLIENTE\_COMPRA\_MODELO se transforma en una tabla intermedia. A esta tabla se le añade la clave primaria propagada de VENDEDOR, ya que es la entidad que tiene relación con la agregación.

## EJEMPLO CON ESPECIALIZACIÓN/GENERALIZACIÓN



- **EMPLEADO**(NUM\_MATRICULA, NOMBRE, DIRECCION, CODIGO\_DEPARTAMENTO)
  - **ADMINISTRATIVO**(NUM\_MATRICULA\_EMPLEADO)
  - **TECNICO**(NUM\_MATRICULA\_EMPLEADO, NIVEL)
  - **DEPARTAMENTO**(CODIGO\_DEPARTAMENTO, NOMBRE\_DEPARTAMENTO)
  - **PROYECTO**(CODIGO\_PROYECTO, NOMBRE\_PROYECTO, FECHA\_INICIO, FECHA\_FIN, CLIENTE)
- ❖ **TECNICO\_TRABAJA\_PROYECTO**(NUM\_MATRICULA\_TECNICO, CODIGO\_PROYECTO, FECHA\_ASIGNACION, FECHA\_CESE)

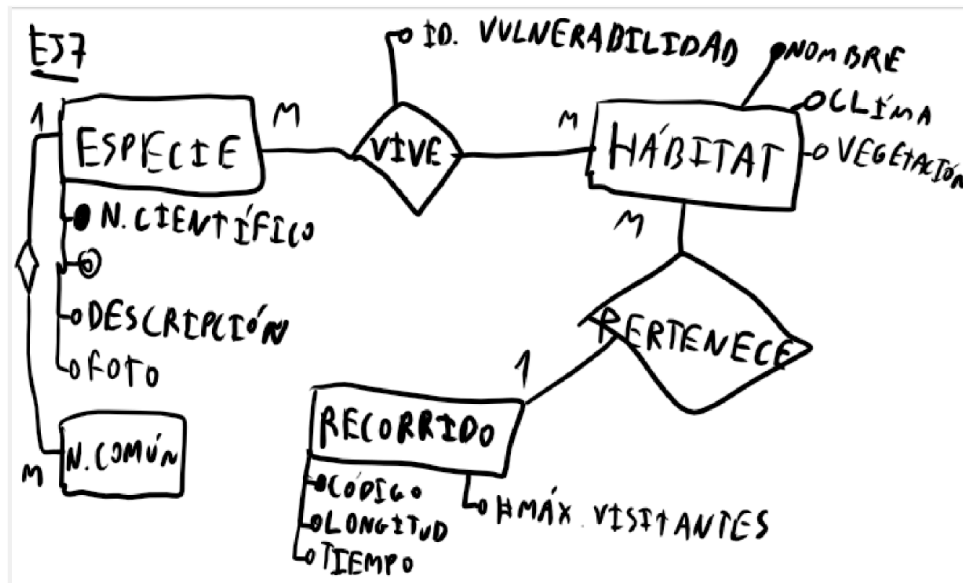
La entidad EMPLEADO se especializa en ADMINISTRATIVO y TECNICO, por lo que las claves primarias de ambos serán la clave primaria propagada de EMPLEADO (añadiendo el nombre de la entidad “padre”, es decir, NUM\_MATRICULA\_EMPLEADO). **Ambas especializaciones deben tener el mismo color de fondo pero cada una debe de tener un color de letra diferente.**

En el caso de la tabla intermedia TECNICO\_TRABAJA\_PROYECTO, se especificará la especialización que se relaciona con PROYECTO. Aquí la clave primaria propagada de TECNICO será NUM\_MATRICULA\_TECNICO (y no NUM\_MATRICULA\_EMPLEADO).

¡OJO! AÑADO ESTOS EJERCICIOS (7 & 11) COMO EJEMPLOS, PERO PODRÍAN CONTENER ERRORES.

#### EJEMPLO CON ENTIDAD SIN ATRIBUTOS DE 1:N

##### EJERCICIO 7



- ESPECIE(NOMBRE CIENTIFICO, DESCRIPCION, FOTO)
- HABITAT(NOMBRE, CLIMA, VEGETACION, CODIGO RECORRIDO)
- RECORRIDO(CODIGO, DURACION, LONGITUD, MAX\_VISITAS)
- ❖ ESPECIE\_VIVE\_HABITAT(NOMBRE CIENTIFICO, NOMBRE HABITAT, VULNERABILIDAD)
- ❖ ESPECIE\_TIENE\_NOMBRE\_COMUN(ID NOMBRE COMUN, NOMBRE CIENTIFICO)

En este ejemplo, NOMBRE\_COMUN no es un atributo (aunque se podría pensar como un atributo multivaluado) ni tampoco se representa como una relación de 1:N normal, sino que se trata de una entidad sin atributos. La clave primaria ID\_NOMBRE\_COMUN la añadimos nosotros junto con una tabla intermedia, ESPECIE\_TIENE\_NOMBRE\_COMUN, que contiene además NOMBRE\_CIENTIFICO como clave propagada de ESPECIE.

## EJEMPLO CON ENTIDAD SIN ATRIBUTOS DE M:N

### EJERCICIO 11



- **PROGRAMA\_VUELO** (COMPañIA, RUTA, CODIGO\_AEROPUERTO\_DESPEGA, CODIGO\_AEROPUERTO\_ATERRIZA)
- **AEROPUERTO** (CODIGO, NOMBRE, CIUDAD, PAIS)
- **VUELO** (VUELO, FECHA, PLAZAS\_VACIAS, COMPañIA\_PROGRAMA\_VUELO, RUTA\_PROGRAMA\_VUELO, NOMBRE\_MODELO\_AVION)
- **MODELO\_AVION** (NOMBRE, CAPACIDAD)

- ❖ **AEROPUERTO\_PUEDE\_ATERRIZAR\_MODELO\_AVION** (CODIGO\_AEROPUERTO, NOMBRE\_MODELO\_AVION)
- ❖ **PROGRAMA\_VUELO\_DIAS** (COMPañIA\_PROGRAMA\_VUELO, RUTA\_PROGRAMA\_VUELO, DIA)