

Práctica 2

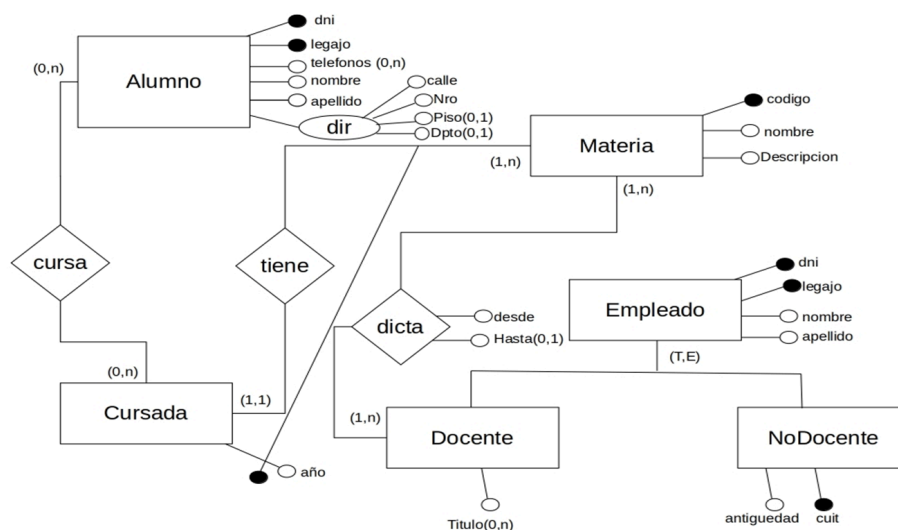
Explicación de Práctica

Modelado Lógico:

El propósito de la generación de un modelo E-R Lógico es convertir el esquema conceptual en un modelo más cercano a la representación entendible por el SGBD. Recordemos que el diseño conceptual busca representar, de la forma más clara posible, las necesidades del usuario. Una vez cumplido este paso, el diseño lógico busca representar un esquema equivalente, que sea más eficiente para su utilización.

Decisiones sobre el Diseño Lógico:

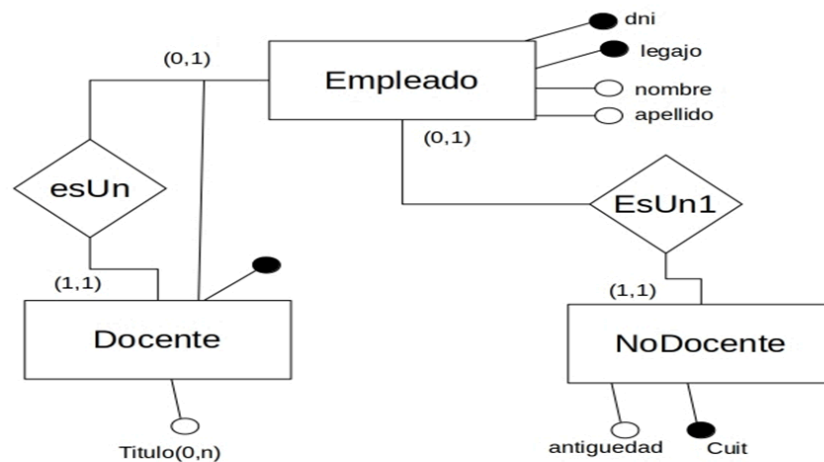
Las decisiones sobre el diseño lógico están vinculadas, básicamente, con cuestiones generales de rendimiento y con un conjunto de reglas que actúan sobre características del esquema conceptual que no están presentes en los SGBD relacionales. Se deben resolver las **jerarquías**, los **atributos compuestos** y los **atributos polivalentes**. Si se parte del **Modelo Conceptual**:



- Resolver las Jerarquías (dependiendo de su cobertura):

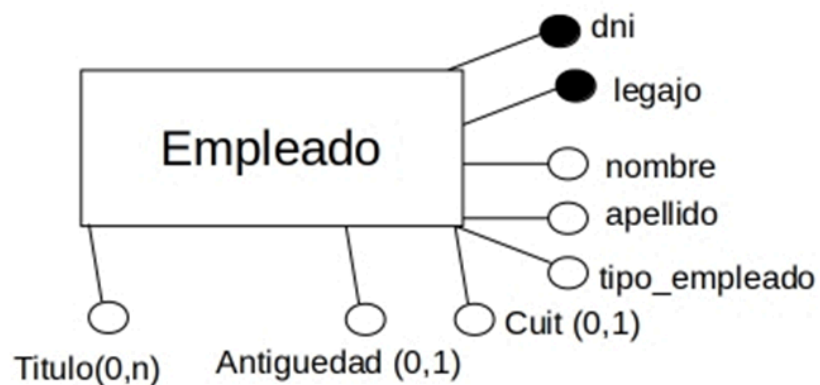
- **Total Exclusiva (T, E)** → Hay 3 posibilidades:

- **Dejar todo:**



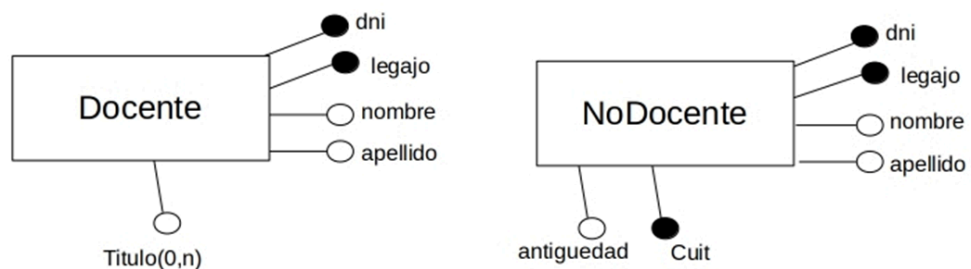
Si las entidades **hijas** no tienen **identificador** debo **bajarlo desde el padre**. Caso contrario es *opcional*. NoDocente puedo no bajarlo, pero si lo bajo no debo cruzarlo con CUIT. (Se dibuja el **identificador externo** tomándolo **desde la línea de la relación no desde la entidad**).

- **Dejar solo al padre:**



Todos los **atributos** de los **hijos** pasan al **padre**. Deben pasar como **no obligatorios**. Lo mismo sucede con las relaciones en los hijos pasan como **relaciones opcionales** (mínima 0). Si en el **hijo** era un **atributo identificador**, **debe dejar de serlo** (nunca un identificador puede ser opcional). Si bien puede deducirse es una **buena opción agregar un atributo que identifique qué tipo de empleado es** (tipo_empleado).

■ **Dejar solo a los hijos:**



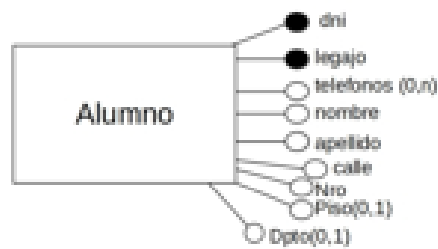
Se deben bajar los **atributos del padre** a **cada uno de los hijos**.

También se deben **bajar las relaciones del padre** (si hubiera).

- **Total Superpuesta (T, S)** → Dos posibilidades, **dejar todo o dejar sólo al padre**. No se puede eliminar al padre.
- **Parcial Exclusiva (P, E)** → Dos posibilidades, **dejar todo o dejar sólo al padre**. No se puede eliminar al padre.
- **Parcial Superpuesta (P, S)** → Dos posibilidades, **dejar todo o dejar sólo al padre**. No se puede eliminar al padre.
- **Resolver Atributos Compuestos:**

Las dos formas más utilizadas para eliminar los atributos compuestos son:

- **Considerar sólo los atributos individuales**

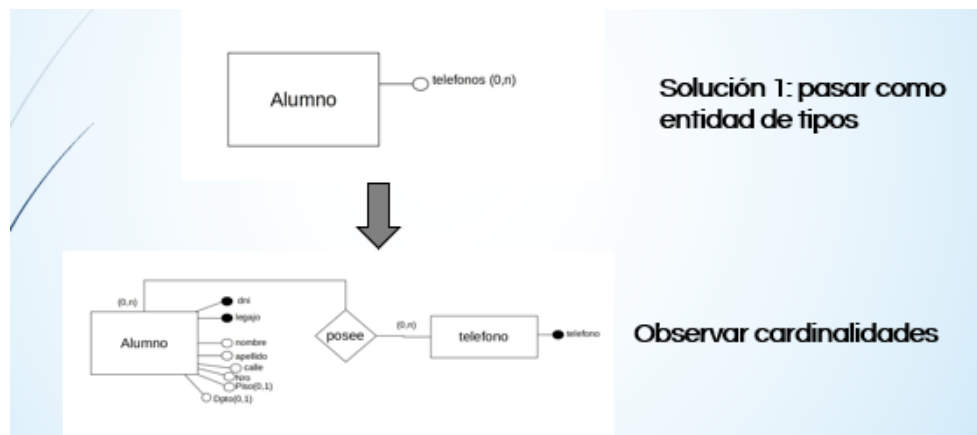


- **Considerar todo en un sólo atributo**

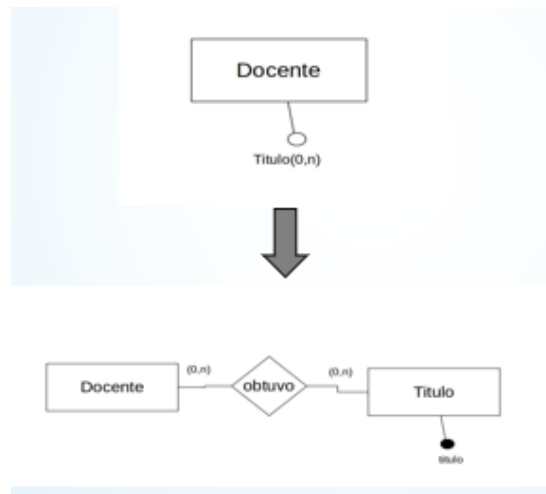


- **Resolver Atributos Polivalentes:**

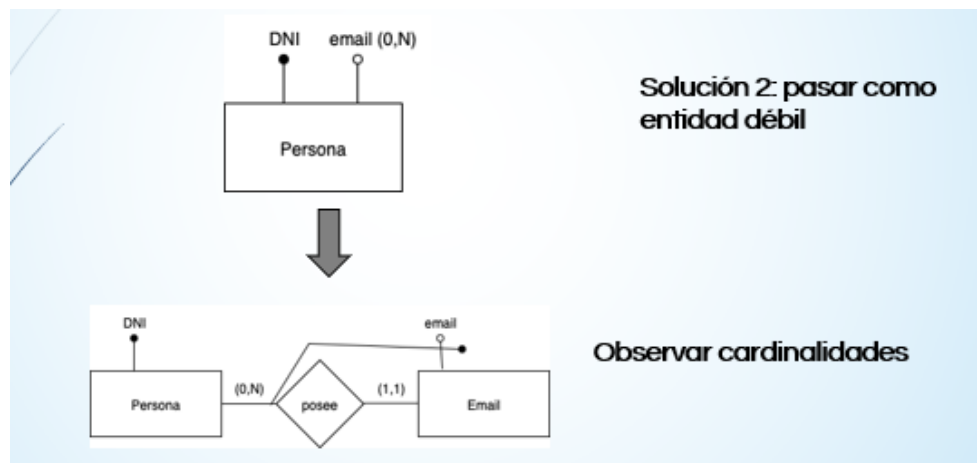
Para resolver los atributos polivalentes se debe **agregar una Entidad y una Interrelación.**



Se pasa del atributo polivalente teléfonos a modelar la Entidad Teléfonos y la Relación Posee para unirlos con Alumnos.

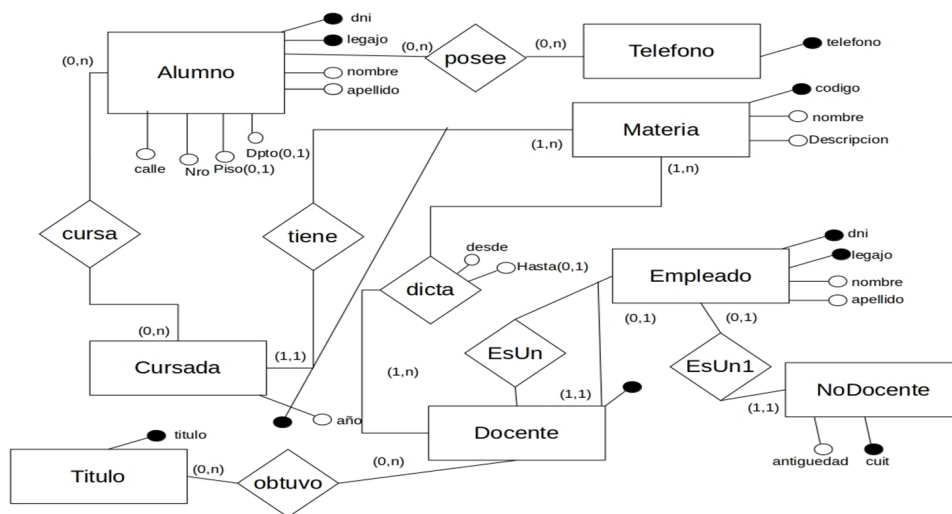


Se pasa del atributo polivalente títulos a modelar la Entidad Títulos y la Relación Obtuvo para unirlos con Docentes.



Esta segunda opción la pasa como una Entidad debil (1,1). Se crea la Entidad Email y la Relación Posee que esta uniendo con Persona. Es entidad debil porque aunque la persona pueda tener muchos emails, cada dirección de email le pertenece solo a una persona.

Modelo Lógico Final



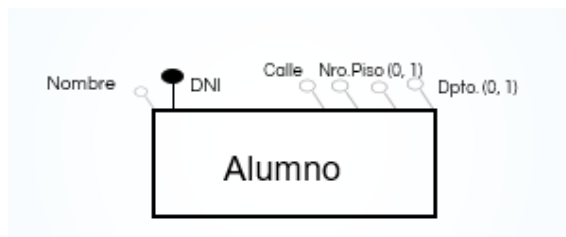
Errores comunes al pasar al modelo lógico:

- No diferenciar correctamente entre *atributos opcionales* y *polivalentes*.
- Convertir *jerarquías* sin considerar el *contexto*.
- Agregar *identificadores innecesarios*.
- Dejar *entidades sin identificar*.
- Modelar incorrectamente los *atributos polivalentes*.

Modelo Físico:

El modelo Físico (relacional) **representa a una BD como una colección de archivos denominados tablas**. Cada tabla se denomina **relación** y está **integrada por filas y columnas**. Cada **fila** se denomina **tupla (registro)** y cada **columna** representa un **atributo**.

Cada **entidad** que se obtuvo en el **modelo lógico**, se transforma en una **tabla**. Por ejemplo:

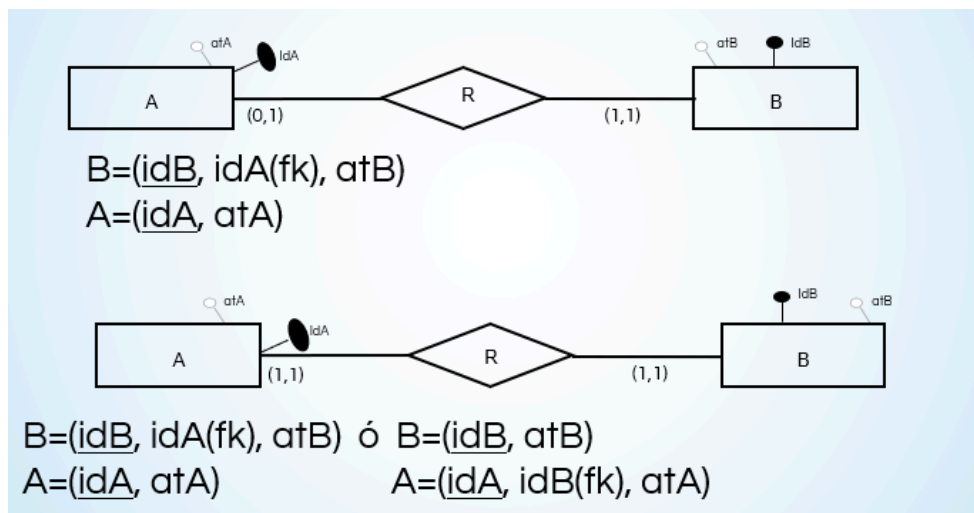


La notación de la tabla de esta entidad es:

Alumno = (DNI, nombre, calle, nro, piso?, depto?)

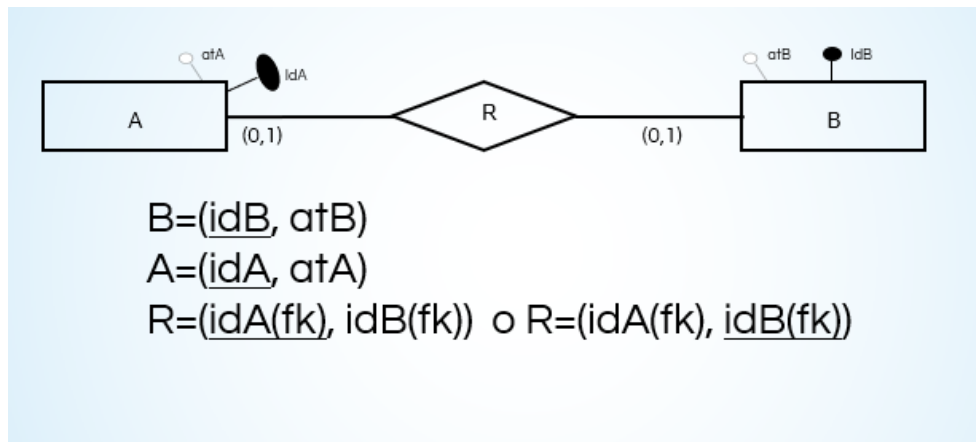
Se marca subrayado el identificador elegido como clave primaria, el resto de los atributos normal, todos separados por comas, y los que tienen ? son aquellos marcados como opcionales.

Una **relación** será una tabla dependiendo de la **cardinalidad** que posee (puede no serlo). Por ejemplo:

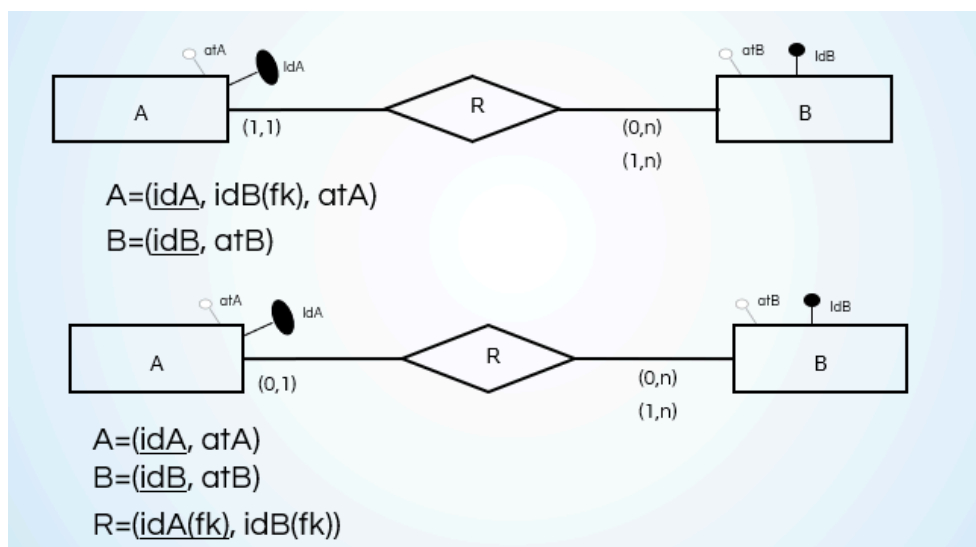


La primera es una relación uno a uno, en este caso la relación **no se convierte en tabla**. Para *mantener el vínculo* con esas dos entidades, paso el identificador de la entidad A a la B porque la B depende de la A (la entidad A puede no estar relacionada con B (0..1) pero no se cumple viceversa). Entonces la entidad B conserva sus atributos y recibe también la **clave foránea** de A que se marca con **(fk)** → **foreign key**.

La segunda es una relación de uno a uno, en este caso **tampoco se convierte la relación en tabla**. Pero en este caso las dos *dependen de si mismas y si o si se relacionan con un elemento de la otra entidad (1..1)*. Por lo que se puede pasar el identificador de A a B o viceversa, siempre que no se pierda el vínculo que genera la relación.

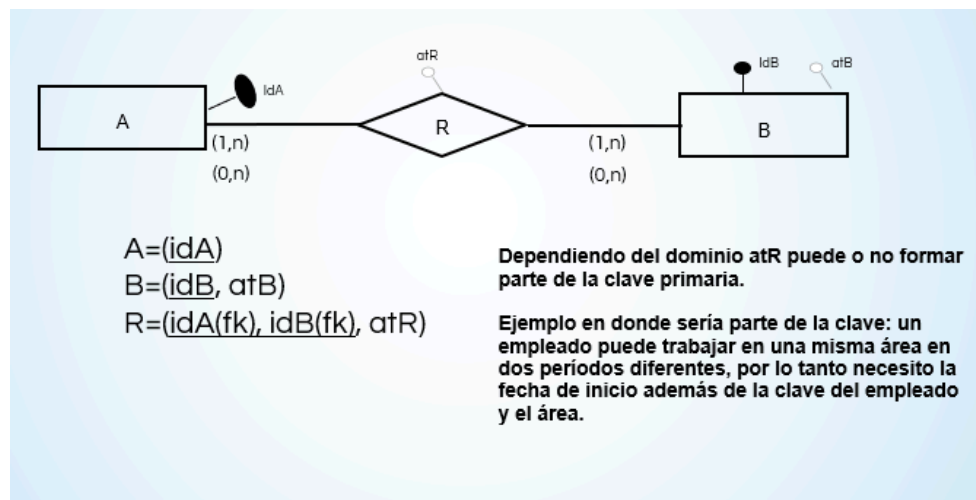


Esta relación tiene una cardinalidad de uno a uno pero opcional de los dos lados (0..1). Como no tengo garantía de que se vayan a relacionar entre sí (puede haber valores nulos en esa conexión) no puedo pasar identificadores de una entidad a otra, estamos **obligados a que la relación se transforme en tabla** (la cual contendrá el identificador de ambas entidades, como se muestra en el gráfico, y puede ser de una manera u otra porque no se van a repetir ninguno de los dos valores).

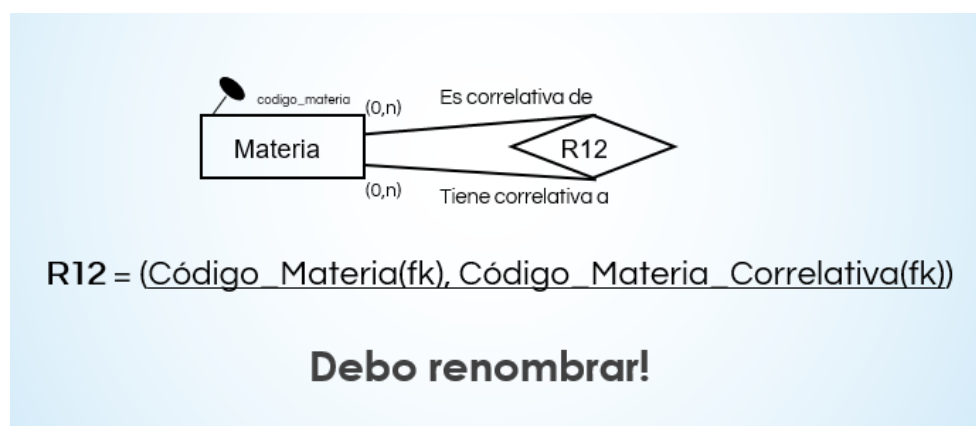


La primera entidad, con cardinalidad uno a muchos, se **evita convertir en tabla**. Se pasa el *identificador de B a la tabla de A* ya que esta última es la *débil*, depende de B y *solo se relacionará con uno de B obligatoriamente*. Si lo hicieramos al revés, pasando el identificador de A a B, podríamos pasar múltiples veces el mismo dato de B a A (porque un elemento de B se puede relacionar con muchos de A diferentes).

El segundo caso también es de uno a muchos pero en este caso se **transforma la relación en tabla** porque ambas entidades pueden o no estar relacionadas (no se podría compartir el identificador de cada entidad con la otra por esa razón). Entonces, se representa la relación con los identificadores de ambas entidades.

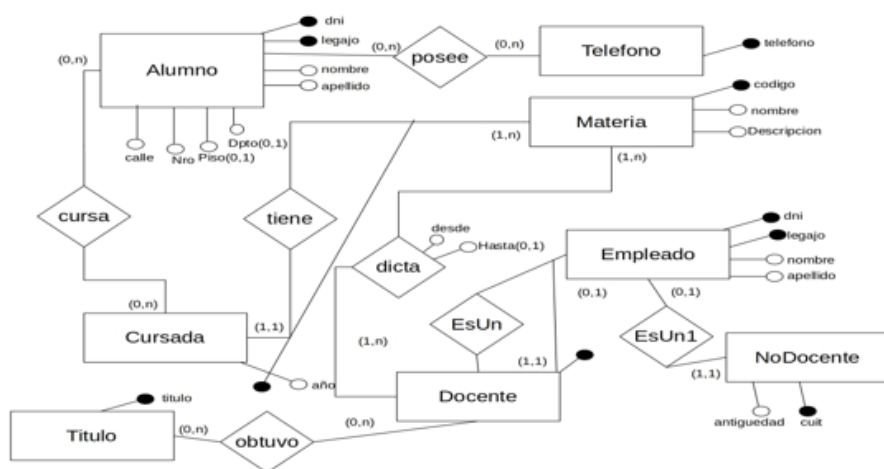


Cuando tenemos una relación de **muchos a muchos**, sin importar si es $(0..n)$ o $(1..n)$, debemos **transformarla en tabla** (evitando redundancia de repetir identificadores en uno y en otro ya que cada elemento de una entidad se puede relacionar con varios de la otra y viceversa). Entonces se modela la tabla que representa a la relación con los identificadores de cada entidad y los atributos que tenga la relación original. El dominio del problema es el que va a determinar si algún atributo de la relación tiene que ser parte de la clave primaria. En la imagen está el ejemplo de un caso donde además de los identificadores se necesitaría agregar a la clave primaria un atributo de la relación.



En una relación recursiva se sigue lo mismo que en las binarias, nada más que hay que tener en cuenta el nombre de los atributos para no repetirlos ya que relacionamos a una entidad consigo misma (estamos obligados a renombrar algún atributo; en el ejemplo se ve que el atributo "código_materia" es el identificador y se pone en la relación con ese nombre pero en lugar de repetirlo 2 veces se renombra en su segunda aparición como "código_materia_correlativa" siendo referencia al mismo atributo).

Ejemplo (pasaje de modelo lógico a físico):



- Alumno = (dni, legajo, nombre, apellido, calle, nro, piso?, dpto?)
- Telefono = (telefono)
- Materia = (codigo, nombre, descripcion)
- Cursada = (año, codigo (fk))
- Empleado = (dni, legajo, nombre, apellido)
- Docente = (dni(fk))
- NoDocente = (cuit, dni(fk), antigüedad)
- Titulo = (titulo)
- obtuvo= (dni(fk), titulo(fk))
- posee = (dni(fk), telefono(fk))
- dicta = (codigo(fk), dni(fk), desde, hasta?)
- curso = (dni(fk), (año, codigo) (fk))

En una tabla no podemos tener dos atributos con el mismo nombre (por ejemplo que se nos cruce DNI original y DNI foráneo).

No vamos a trabajar con las claves/identificadores autoincrementales en la práctica (lo que vimos en la teoría).

Errores típicos:

- No marcar **claves primarias** en alguna tabla.
- Indicar una clave primaria compuesta porque había **dos posibles identificadores** en vez de **elegir uno**.
- Introducir un campo "**id**" de forma **innecesaria**.
- Resolver incorrectamente las **relaciones** que tienen **cardinalidad opcional** en algún lado.
- Marcar claves primarias no mínimas.
- No subrayar con una línea continua la **PK** en caso de ser **compuesta**.
- No marcar las **FK** en las **tablas** (tanto las derivadas de **entidades** como de **relaciones**).
- Olvidarse **convertir** alguna **entidad o relación**.
- En jerarquías ya resueltas, trasladar *todos los atributos* de la tabla *padre a las hijas* en lugar de *solo su clave primaria*.

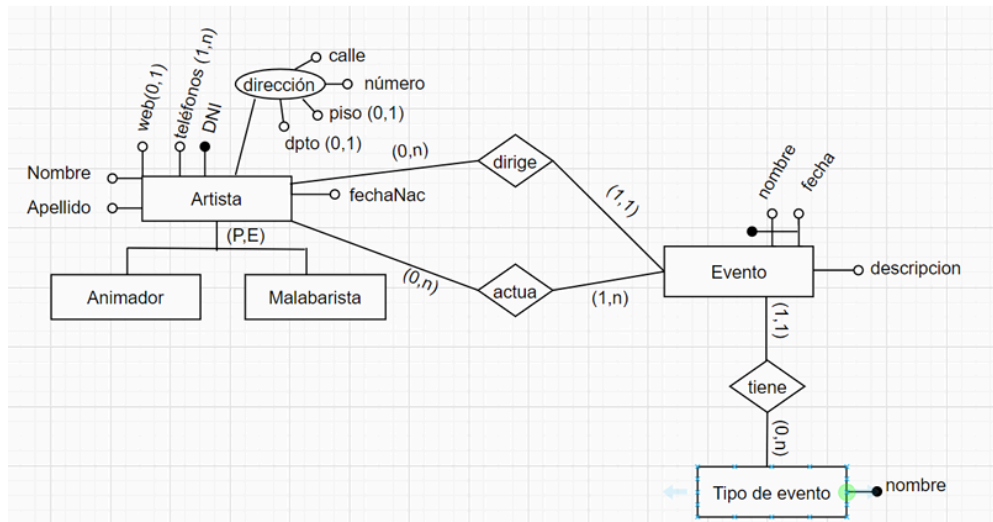
Ejercicios

Derivación a Modelos Lógico y Físico:

Para cada ejercicio plantear el correspondiente **pasaje al modelo lógico y al modelo físico**.

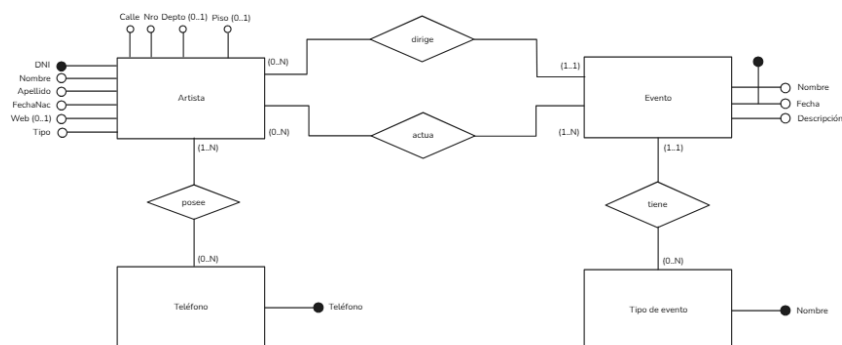
Convenciones para el **modelo físico**:

- *Subrayar con una única línea las claves primarias.*
- Denotar como **fk** a las *claves externas*.
- Denotar *atributos opcionales* con el **signo de interrogación** (por ejemplo, b3?).



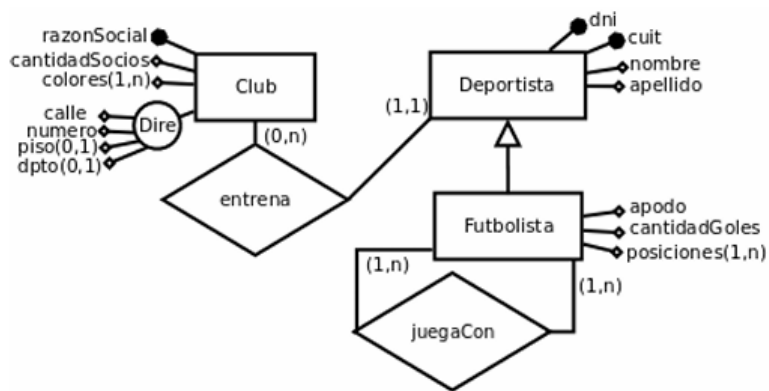
1.

Modelo lógico:



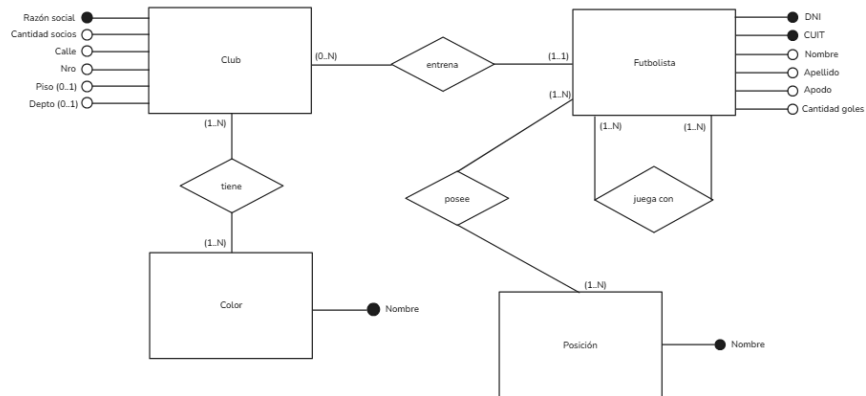
Modelo físico:

- **Artista** = (DNI, nombre, apellido, fechanac, tipo, calle, nro, depto?, piso?, web?)
- **Teléfono** = (teléfono)
- **Evento** = (nombre, fecha, descripción, tipo_evento(fk), DNI_director(fk))
- **TipoDeEvento** = (nombre)
- **posee** = (DNI(fk), teléfono(fk))
- **actua** = (DNI(fk), (nombre, fecha)(fk))



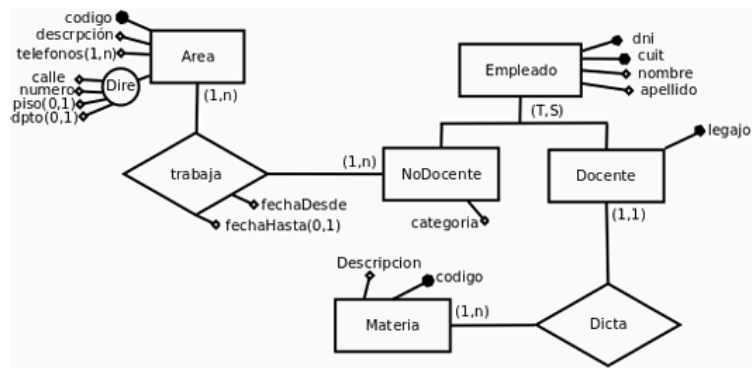
2.

Modelo lógico:



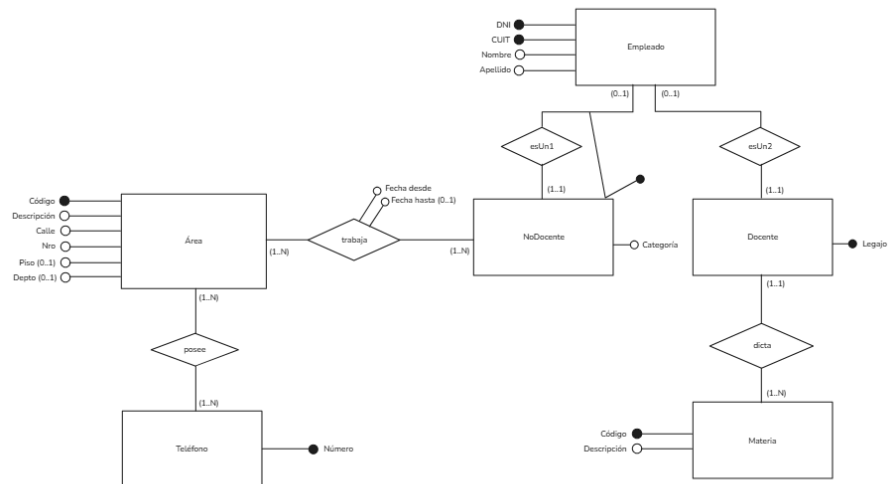
Modelo físico:

- **Club** = (razónSocial, cantidadSocios, calle, nro, piso?, depto?)
- **Color** = (nombre)
- **Posición** = (nombre)
- **Futbolista** = (DNI, CUIT, nombre, apellido, apodo, cantidadGoles, razónSocial(fk))
- **tiene** = (razónSocial(fk), nombre(fk))
- **posee** = (DNI(fk), nombre(fk))
- **juegaCon** = (DNI1(fk), DNI2(fk))



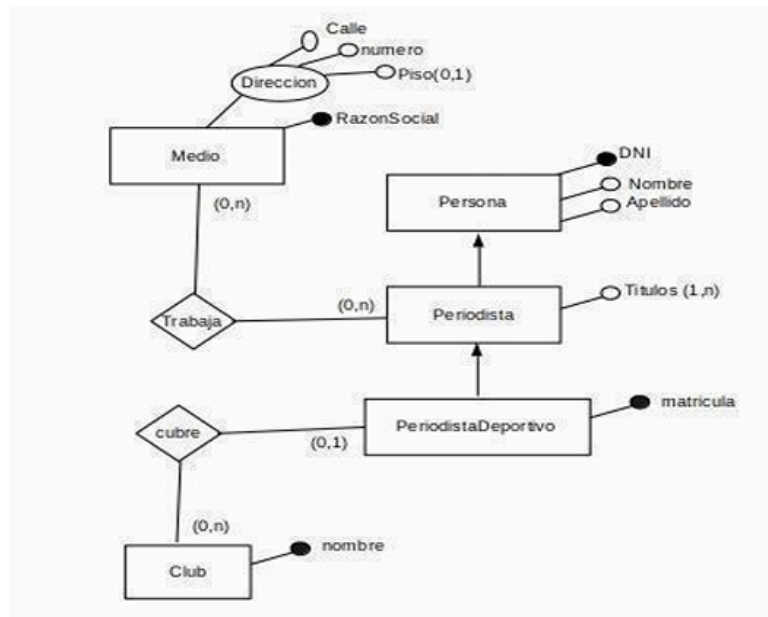
3.

Modelo lógico:



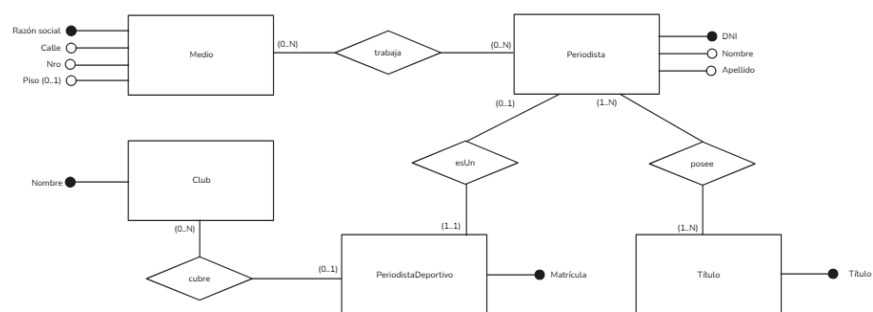
Modelo físico:

- **Área** = (código, descripción, calle, nro, piso?, depto?)
- **Teléfono** = (número)
- **Empleado** = (DNI, CUIT, nombre, apellido)
- **Docente** = (legajo, código(fk), DNI(fk))
- **NoDocente** = (DNI(fk), categoría)
- **Materia** = (código, descripción)
- **posee** = (código(fk), número(fk))
- **trabaja** = (código(fk), DNI(fk), fechaDesde, fechaHasta?)



4.

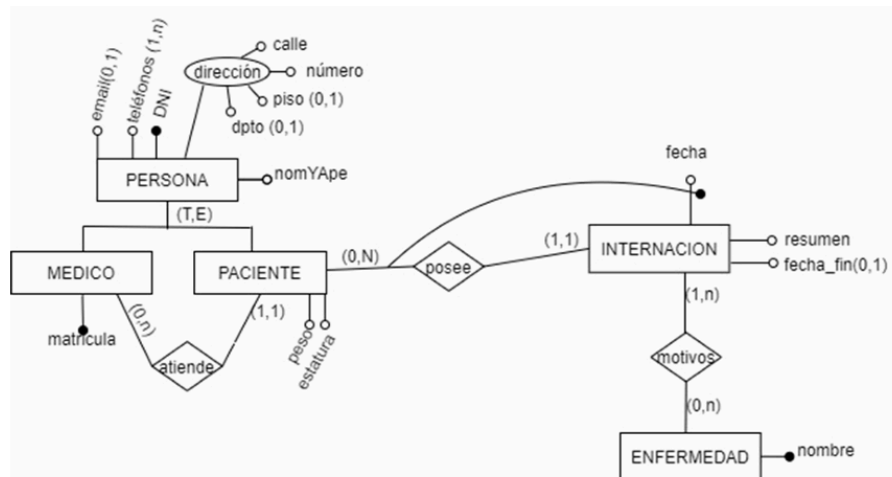
Modelo lógico:



Modelo físico:

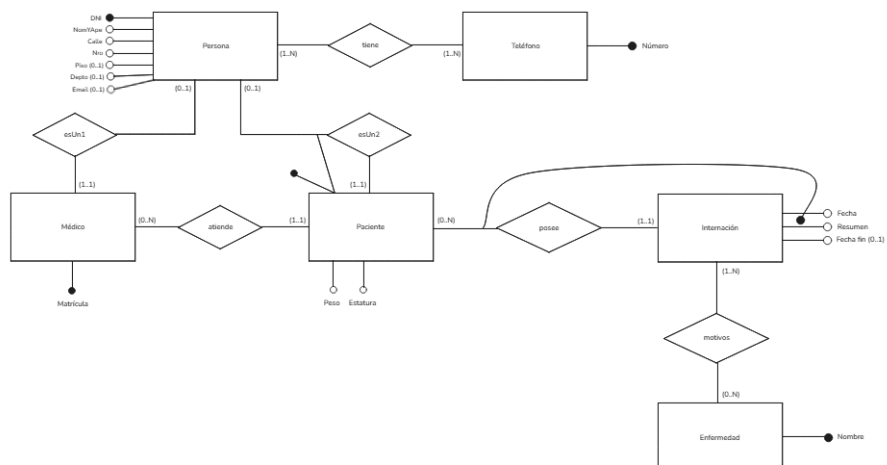
- **Medio** = (razónSocial, calle, nro, piso?)
- **Club** = (nombre)
- **Periodista** = (DNI, nombre, apellido)
- **PeriodistaDeportivo** = (matrícula, DNI(fk))
- **Título** = (título)

- **trabaja** = (razónSocial(fk), DNI(fk))
- **posee** = (DNI(fk), título(fk))
- **ubre** = (matrícula(fk), nombre(fk))



5.

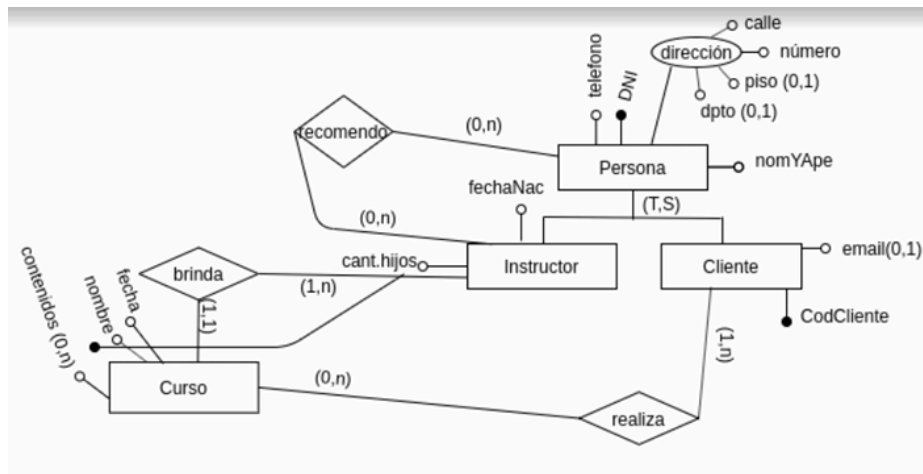
Modelado lógico:



Modelado físico:

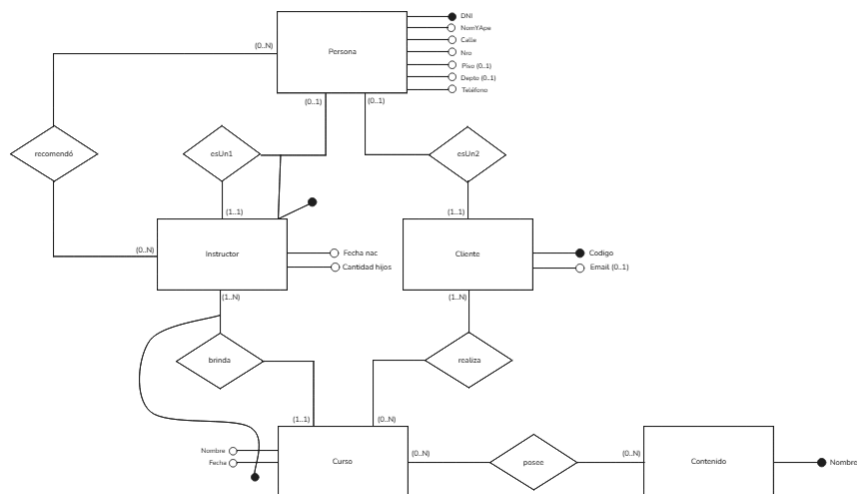
- **Persona** = (DNI, nomYApe, calle, nro, piso?, depto?, email?)
- **Médico** = (matrícula, DNI(fk))
- **Paciente** = (DNI(fk), peso, estatura, matrícula(fk))
- **Teléfono** = (número)

- **Internación** = (fecha, DNI(fk), resumen, fechaFin?)
- **Enfermedad** = (nombre)
- **tiene** = (DNI(fk), número(fk))
- **motivos** = (((DNI, fecha)(fk), nombre(fk)))



6.

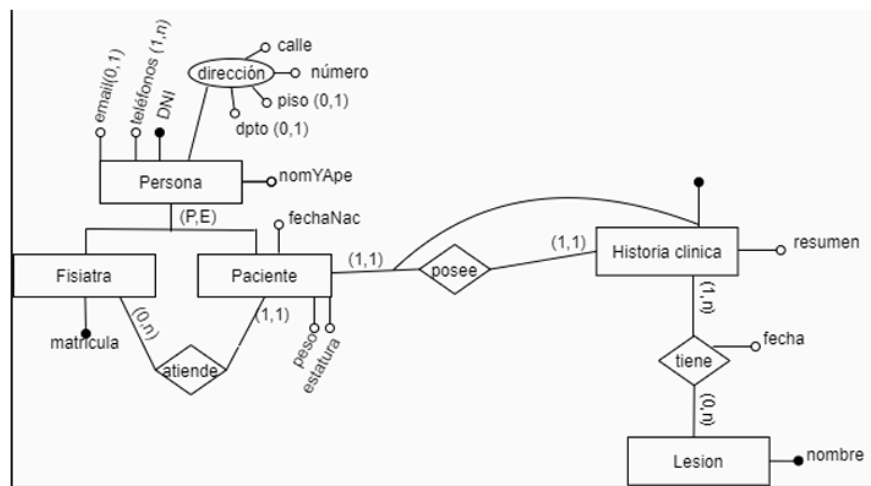
Modelo lógico:



Modelo físico:

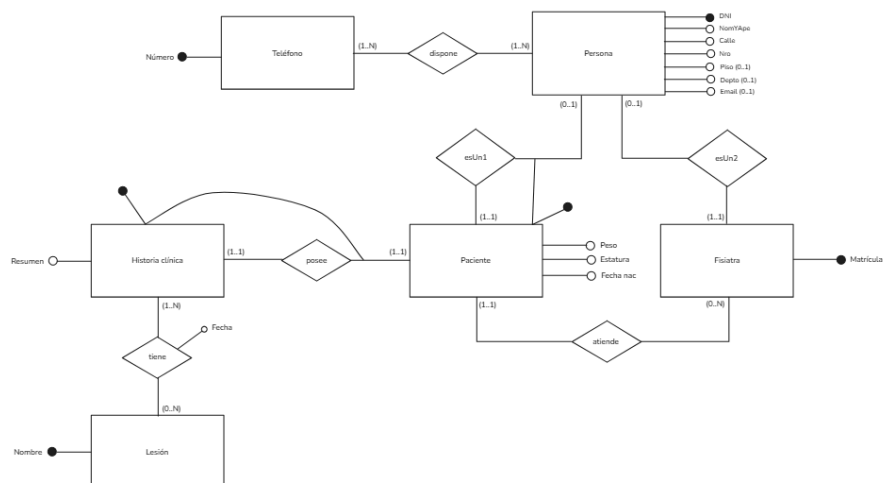
- **Persona** = (DNI, nomYApe, calle, nro, piso?, depto?, teléfono)
- **Instructor** = (DNI(fk), fechaNac, cantidadHijos)

- **Cliente** = (codigo, email?, DNI(fk))
- **Curso** = (DNI(fk), nombre, fecha)
- **Contenido** = (nombre)
- **recomendo** = (DNI_persona(fk), DNI_instructor(fk))
- **realiza** = (codigo(fk), (DNI, nombre, fecha)(fk))
- **posee** = ((DNI, nombre, fecha)(fk), nombre_contenido(fk))



7.

Modelo lógico:



Modelo físico:

- **Persona** = (DNI, nomYape, calle, nro, piso?, depto?, email?)
- **Paciente** = (DNI(fk), peso, estatura, fechaNac, matrícula(fk))
- **Fisiatra** = (matrícula, DNI(fk))
- **Teléfono** = (número)
- **HistoriaClínica** = (DNI(fk), resumen)
- **Lesión** = (nombre)
- **dispone** = (número(fk), DNI(fk))
- **tiene** = (DNI(fk), nombre(fk), fecha)