

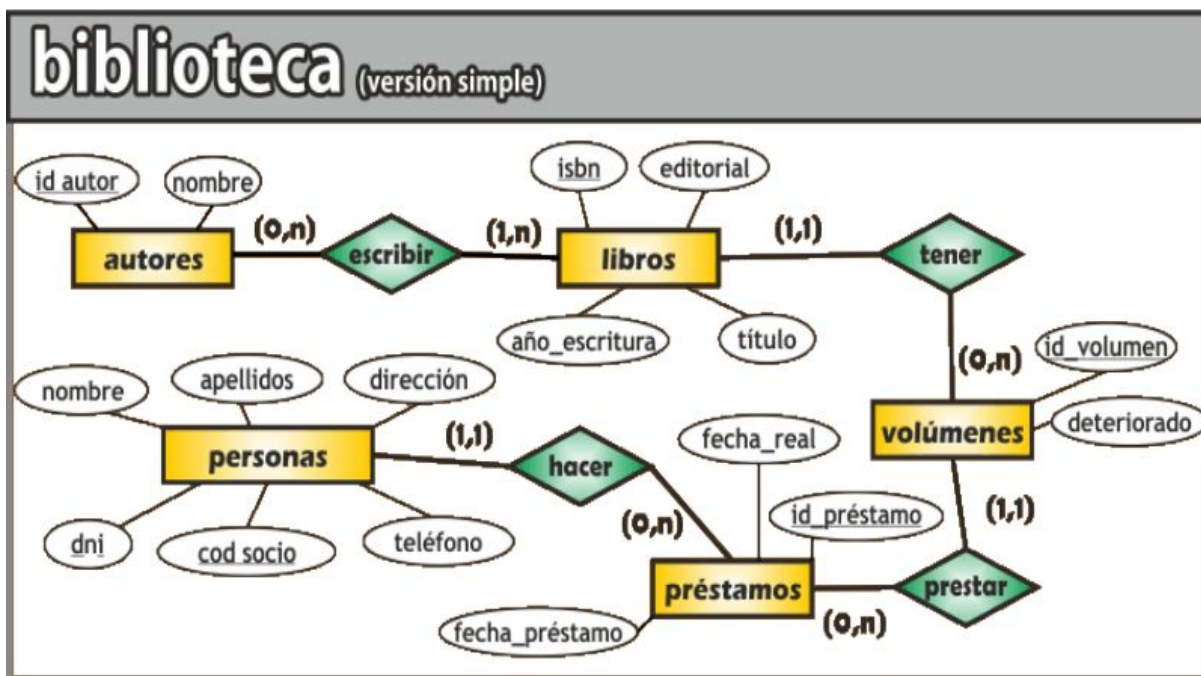
# EJERCICIOS E/R

## Ejercicio 1. Biblioteca v1

Crear un diseño entidad relación (estando prohibido utilizar símbolos del modelo extendido) que permita gestionar los datos de una biblioteca de modo que:

- Las personas socias de la biblioteca disponen de un código de socio y además necesitan almacenar su dni, dirección, teléfono, nombre y apellidos
- La biblioteca almacena libros que presta a los socios y socias, de ellos se almacena su título, su editorial, el año en el que se escribió el libro, el nombre completo del autor (o autores), el año en que se editó y en qué editorial fue y el ISBN.
- Necesitamos poder indicar si un volumen en la biblioteca está deteriorado o no
- Queremos controlar cada préstamo que se realiza almacenando la fecha en la que se realiza, la fecha tope para devolver (que son 15 días más que la fecha en la que se realiza el préstamo) y la fecha real en la que se devuelve el libro.

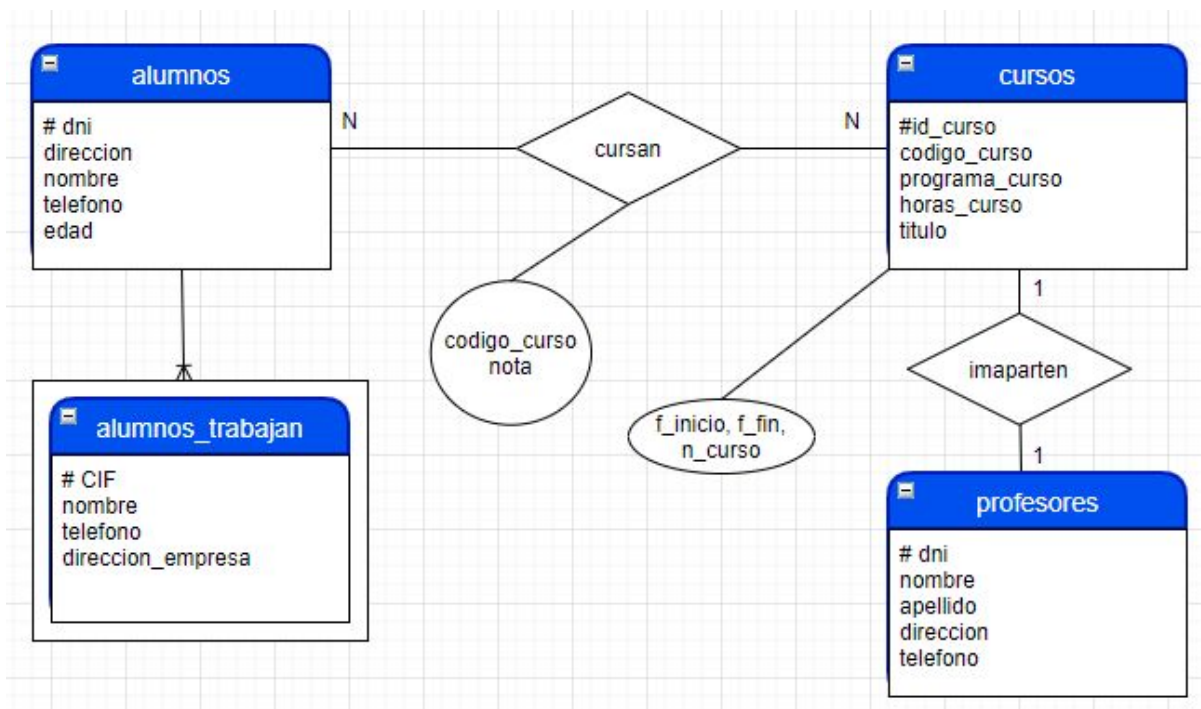
## 4. Representación Gráfica.



## Ejercicio 2. Academia de clases

Crear un diseño entidad relación que permita controlar el sistema de información de una academia de cursos siguiendo estas premisas:

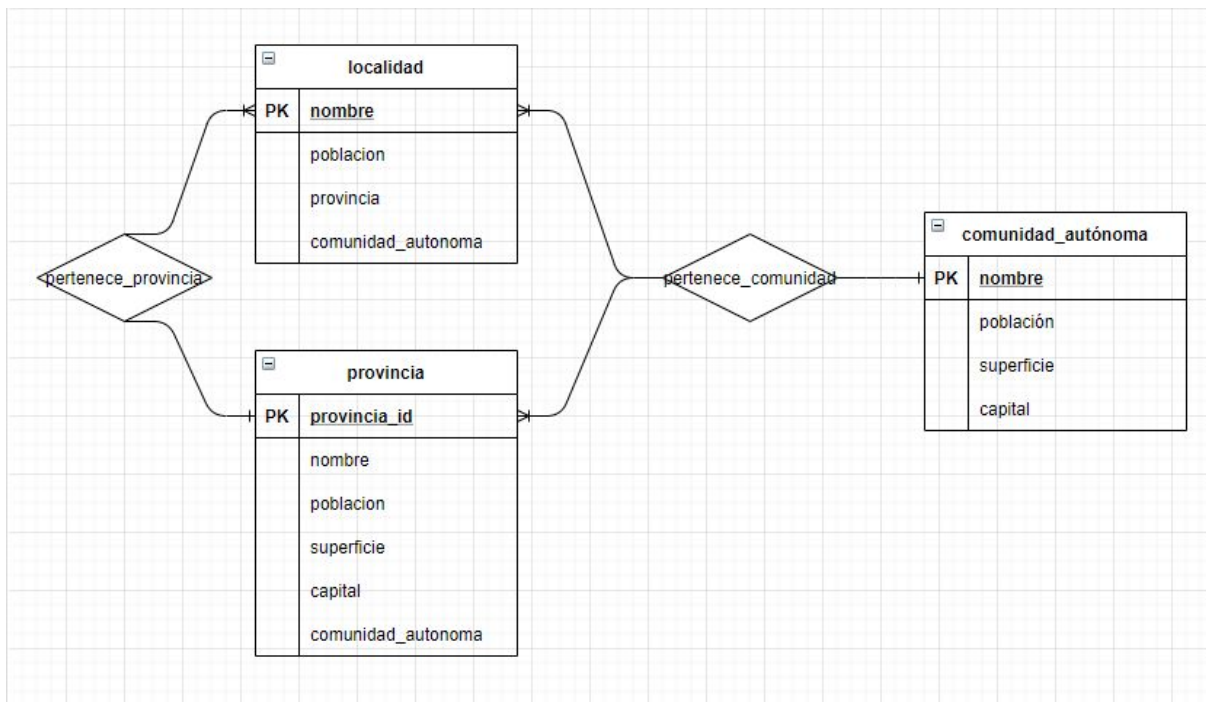
- Se dan clases a trabajadores y desempleados. Los datos que se almacenan de los alumnos son el DNI, dirección, nombre, teléfono y la edad
- Además de los que trabajan necesitamos saber el CIF, nombre, teléfono y dirección de la empresa en la que trabajan
- Los cursos que imparte la academia se identifican con un código de curso. Además se almacena el programa del curso, las horas de duración del mismo, el título y cada vez que se imparte se anotará las fechas de inicio y fin del curso junto con un número concreto de curso (distinto del código) y los datos del profesor o profesora (sólo uno por curso) que son: dni, nombre, apellidos, dirección y teléfono
- Se almacena la nota obtenida por cada alumno en cada curso teniendo en cuenta que un mismo alumno o alumna puede realizar varios cursos y en cada cual obtendrá una nota.



## Ejercicio 3. Geografía

Crear un diseño entidad relación que permita almacenar datos geográficos referidos a España:

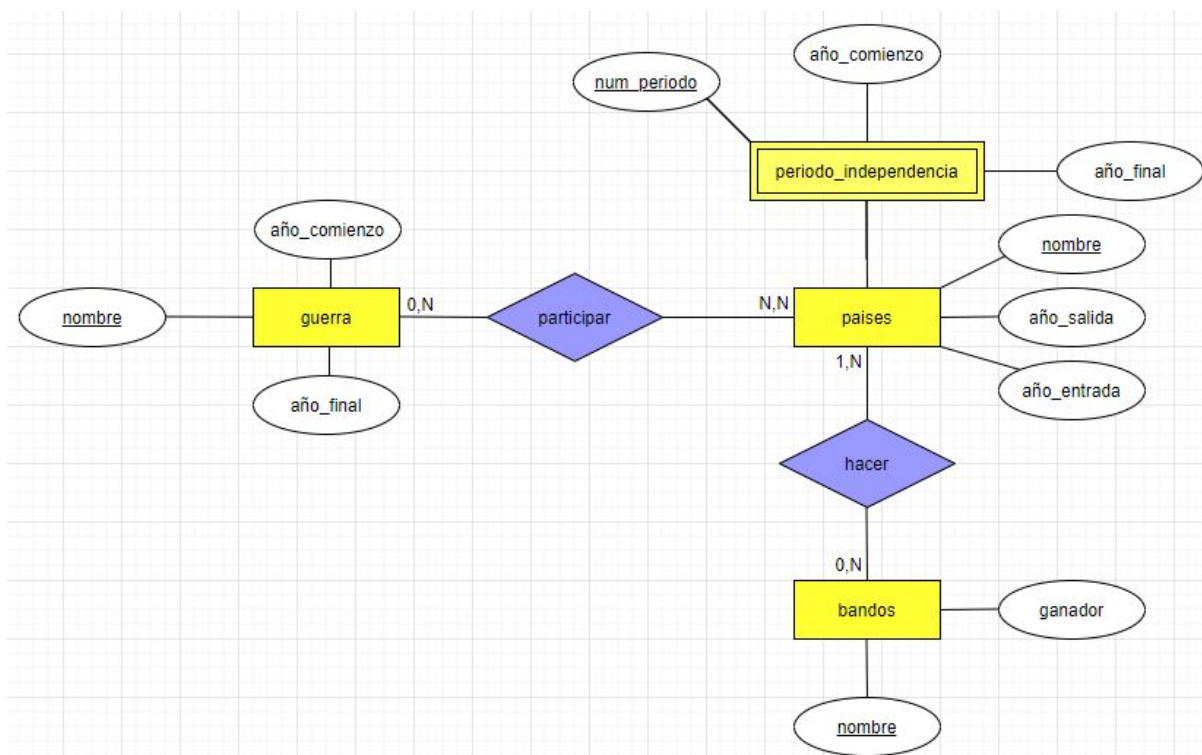
- Se almacenará el nombre y población de cada localidad, junto con su nombre y los datos de la provincia a la que pertenece la localidad, su nombre, población y superficie.
- Necesitamos también conocer los datos de cada comunidad autónoma, nombre, población y superficie y por supuesto las localidades y provincias de la misma
- Para identificar a la provincia se usarán los dos primeros dígitos del código postal. Es decir 34 será el código de Palencia y 28 el de Madrid
- Necesitamos saber qué localidad es la capital de cada provincia y cuáles lo son de cada comunidad.



## Ejercicio 4. Guerras

Diseñar un modelo entidad/relación que almacene los datos de todas las guerras de la historia de modo que:

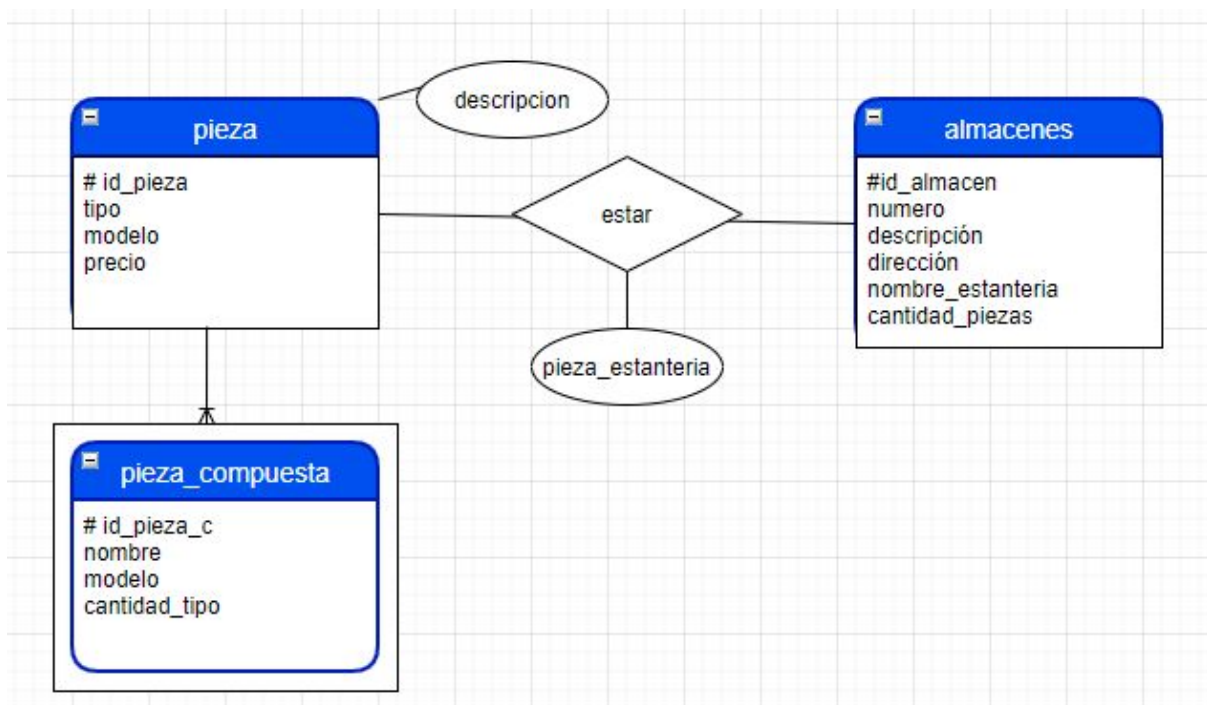
- Se almacene el año en el que empezó la guerra y el año en que terminó, así como su nombre y el de los países contendientes, pudiendo indicar además quienes fueron los ganadores
- Hay que tener en cuenta que los países se pueden unir a la guerra a uno u otro bando (suponemos que solo hay dos bandos) después de comenzada la guerra (como EEUU en la 2a guerra mundial) y que incluso pueden abandonar la guerra antes de que esta finalice (como Rusia en la 1a guerra mundial)
- Los países que se almacenan en la base de datos pueden no ser países actualmente (como Prusia, Aragón, Asiria, etc.) por lo que se ha contemplado que en la base de datos se almacenen los años en los que el país ha sido independiente, teniendo en cuenta que hay países que ha habido momentos en los que ha sido independiente y otros en los que no (por ejemplo Croacia). Bastará con almacenar los periodos en los que ha sido independiente.



## Ejercicio 5. Almacén v1

Se trata de crear una base de datos sobre un almacén de piezas de modo que:

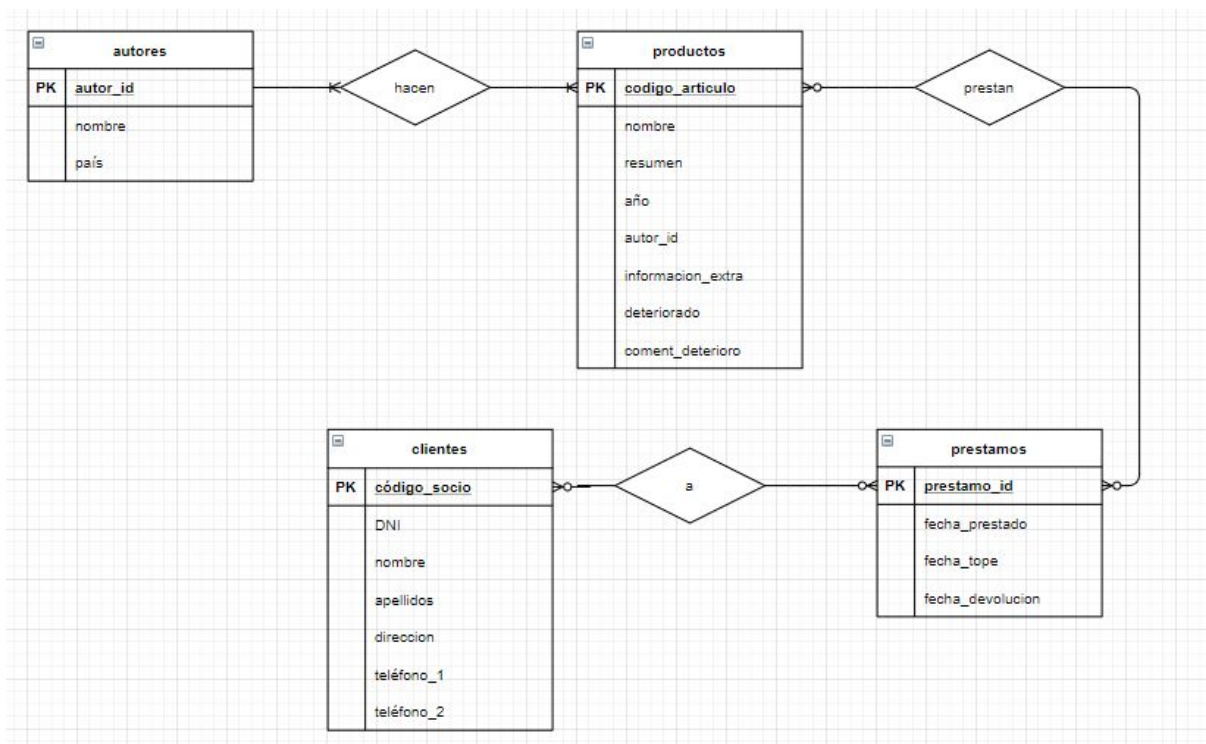
- Cada pieza se identifica con dos letras (tipo, por ejemplo *TU*=tuerca) y un número (modelo, por ejemplo 6)
- Almacenamos un atributo que permite saber la descripción de cada tipo de pieza. Es decir el tipo *TU* tendrá la descripción *tuerca*.
- Necesitamos conocer el precio al que vendemos cada pieza.
- Además hay piezas que se componen de otras piezas, por ejemplo una puerta se compone de una hoja de madera, una bisagra y un picaporte. Incluso una pieza puede estar compuesta de otras piezas que a su vez pueden estar compuestas por otras y así sucesivamente
- Tenemos una serie de almacenes de los que guardamos su número, descripción, dirección y el nombre de cada estantería de almacén. Cada estantería se identifica por tres letras.
- Necesitaremos saber la cantidad de piezas que tenemos en cada almacén y saber en qué estanterías están las piezas buscadas.



## Ejercicio 6. Biblioteca v2

Se trata de crear una base de datos sobre el funcionamiento de una biblioteca

- Almacenaremos el DNI, nombre, apellidos, código de socio, dirección y teléfonos (pueden ser varios, pero al menos uno)
- La biblioteca presta libros, CDs y películas. De todos ellos se almacena un código de artículo distinto para cada pieza en la biblioteca. Es decir si tenemos tres libros del Quijote, los tres tendrán un número distinto de artículo.
- Además almacenamos el nombre de cada artículo, el año en el que se hizo la obra (sea del tipo que sea) un resumen de la obra y los datos de los autores del mismo. Se considera autor de la película al director, de la música al intérprete y del libro al escritor. Pero de todos ellos se guarda la misma información: nombre y país.
- De los libros además se guarda el número de páginas, de los CDs el número de canciones y de la película la duración
- Anotamos si un artículo concreto está deteriorado y un comentario sobre el posible deterioro
- Cuando se presta un artículo, se anota fecha en la que se presta y la fecha tope para devolverle. Cuando el socio le devuelve, se anota la fecha de devolución.
- No hay tope sobre el número de artículos que puede prestarse a un socio e incluso el socio podría llevarse varias veces el mismo artículo en distintos préstamos.

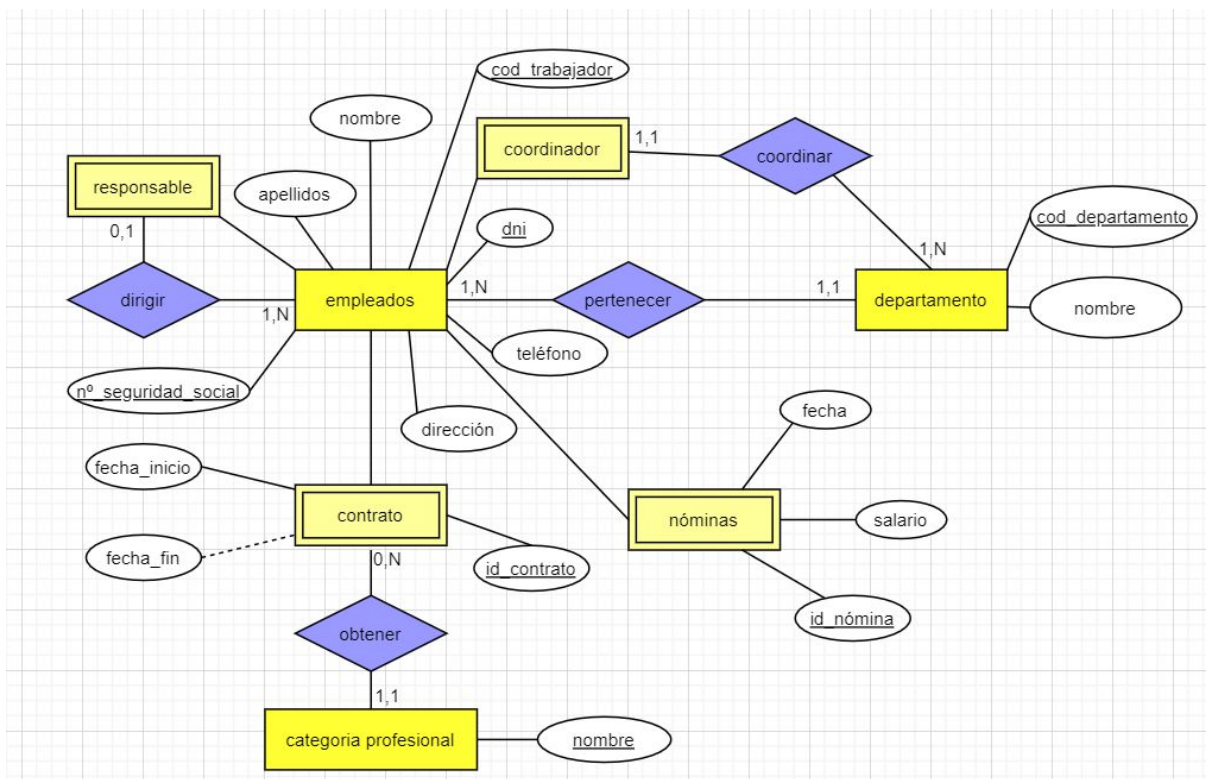




## Ejercicio 7. Organigrama

Crear el esquema entidad/relación que represente el organigrama de una empresa, de modo que:

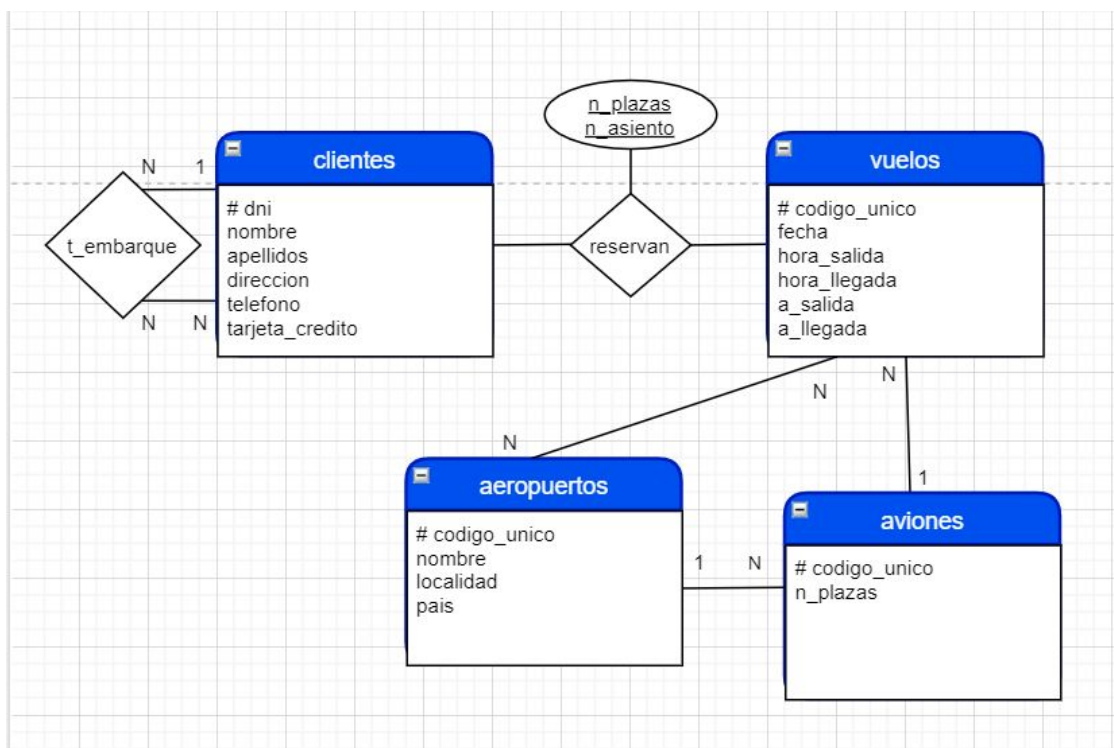
- Aparezcan los datos de todos los empleados y empleadas: dni, no de seguridad social, código de trabajador, nombre, apellidos, dirección, teléfono y departamento en el que trabajan indicado por su código y nombre.
- También hay que tener en cuenta que cada trabajador puede tener un responsable (que en realidad es otro trabajador)
- Los departamentos poseen un único coordinador del mismo
- Necesitamos almacenar la categoría profesional de los trabajadores y trabajadoras, teniendo en cuenta que la categoría a veces cambia al cambiar el contrato, de los contratos se almacena la fecha de inicio del mismo y la fecha final (un contrato en vigor tendrá como fecha final el valor nulo).
- También controlaremos las nóminas que ha recibido el trabajador de las que sabemos la fecha, el salario y a qué trabajador van dirigidas y la categoría del mismo.



## Ejercicio 8. Vuelos

Crear el esquema entidad/relación que permita gestionar reservas de vuelos, de modo que:

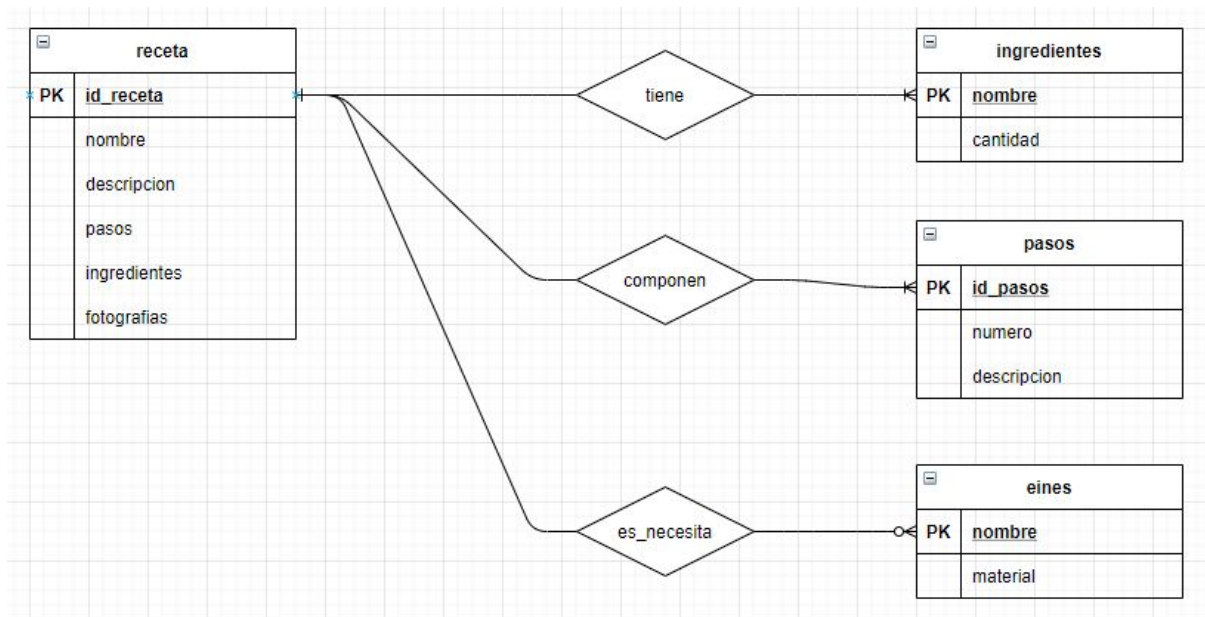
- Los clientes pueden reservar vuelos. Con la reserva se pueden reservar varias plazas, pero no poseeremos el número de asiento hasta obtener la tarjeta de embarque. En ese instante se asignará el asiento que tiene como identificación la fila, columna y la planta en la que está situado.
- Se pueden obtener tarjetas de embarque sin tener reserva
- Las tarjetas de embarque se refieren a un único cliente. De modo que aunque reservemos nueve plazas, cada cliente podrá sacar su tarjeta de embarque indicando el número de reserva, la fecha de la misma y sus datos personales (dni, nombre, apellidos, dirección y teléfono). Además la persona que reserva debe indicar una tarjeta de crédito que quedará asociada a esa persona.
- El vuelo que se reserva tiene un código único, una fecha y una hora de salida y de llegada y un aeropuerto de salida y otro de llegada
- Los aeropuertos poseen un código único, además del nombre y la localidad y el país en el que se encuentran
- Se guarda información sobre los aviones, código y número de plazas. Los vuelos sólo les puede realizar un avión determinado, pero el mismo avión puede realizar (como es lógico) otros vuelos.





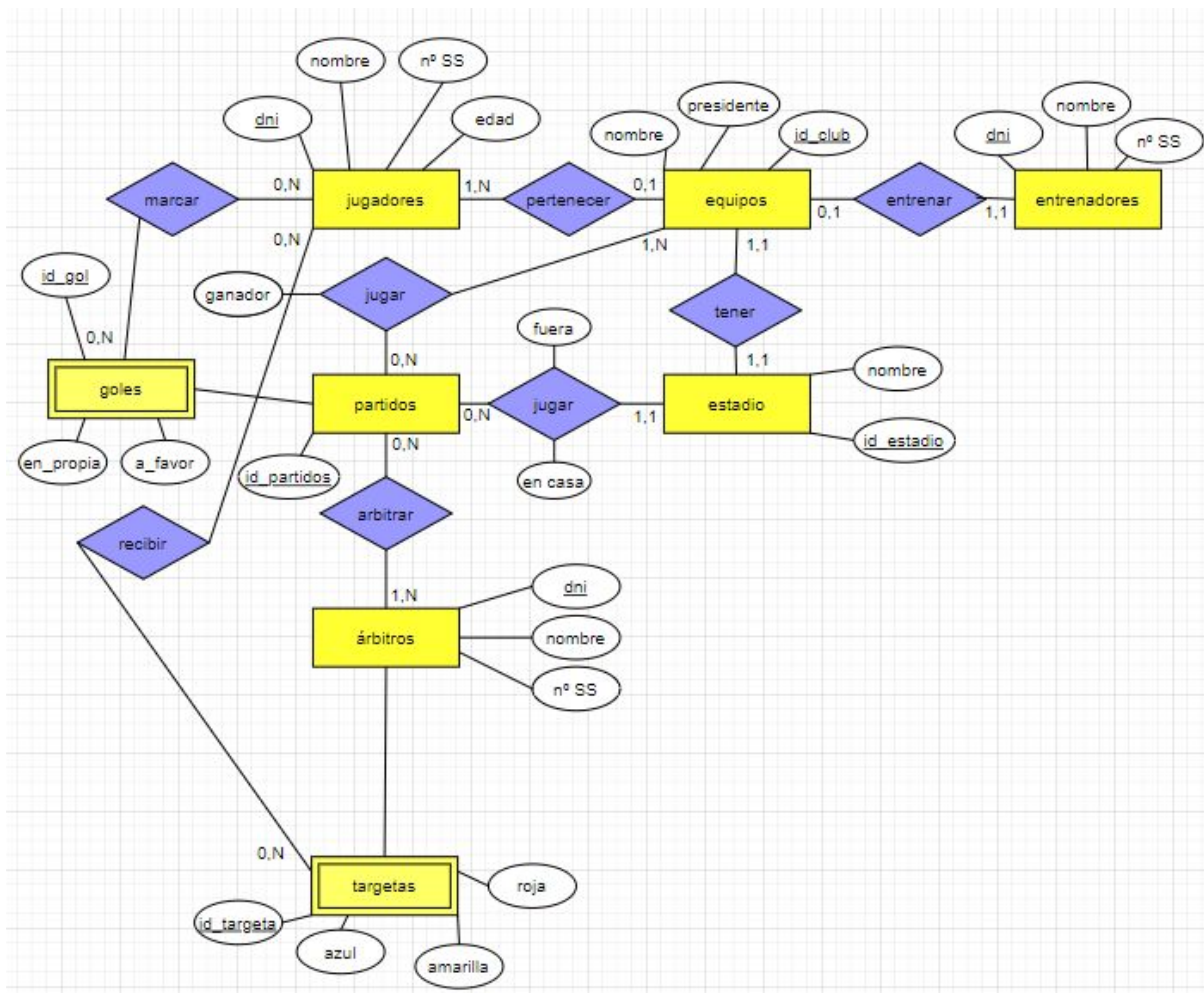
## Ejercicio 9. Recetas de cocina

Crear el esquema entidad/relación que permita gestionar los datos sobre preparación de recetas de cocina.



## Ejercicio 10. Futbol

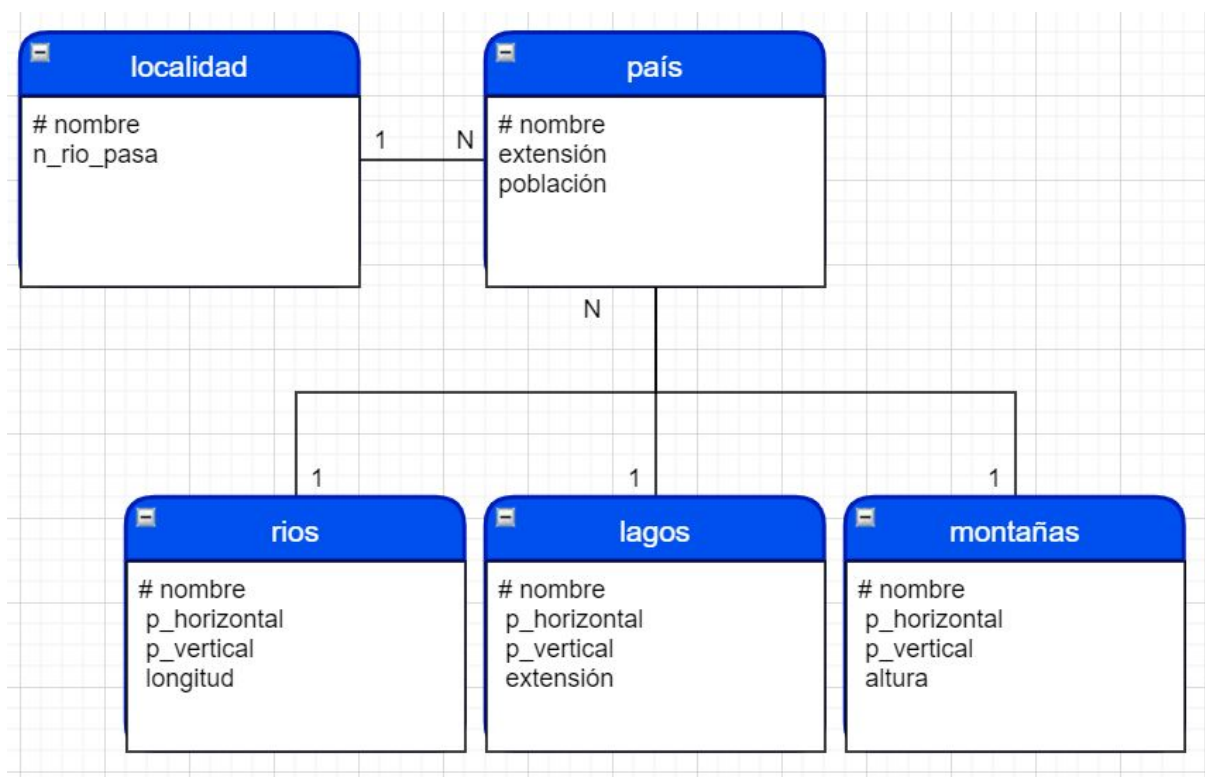
Crear el esquema entidad/relación que permita crear el diseño de una base de datos que almacena información sobre los partidos de una liga de futbol una temporada. Hay que tener en cuenta que en dicha liga los jugadores no pueden cambiar de equipo.



## Ejercicio 11. Accidentes geográficos

Realizar un esquema entidad/relación que sirva para almacenar información geográfica. Para ello hay que tener en cuenta

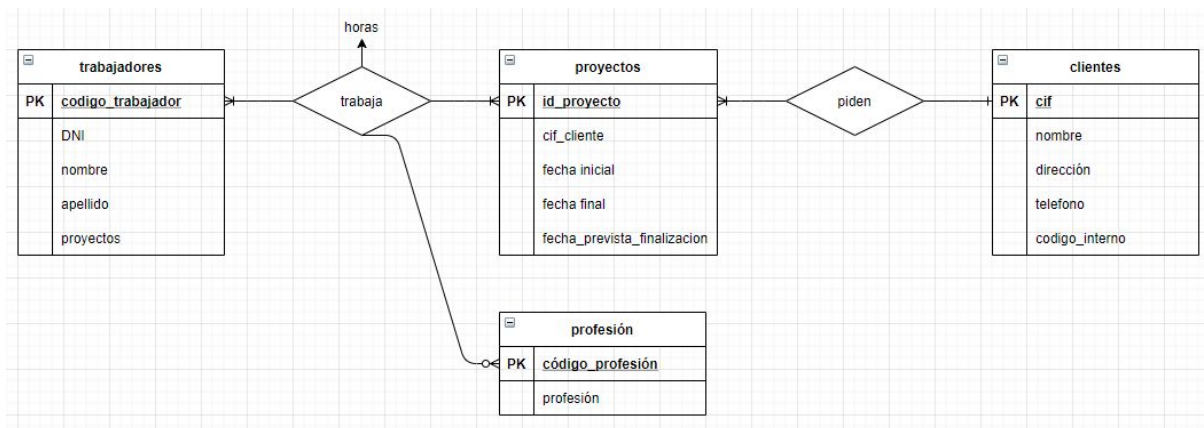
- Se almacenan los siguientes accidentes geográficos: ríos, lagos y montañas
- De cada accidente se almacenan su posición horizontal y vertical según el eje de la tierra, además de su nombre
- De los ríos se almacena su longitud, de las montañas su altura y de los lagos su extensión
- Se almacena también información sobre cada país, su nombre, su extensión y su población
- Se desea almacenar información que permite saber en qué país está cada accidente geográfico, teniendo en cuenta que cada accidente puede estar en más de un país.
- Se almacena también los nombres de cada localidad del planeta. Y se almacena por qué localidades pasa cada río.



## Ejercicio 12. Empresa de software

Realizar un esquema entidad/relación que permita modelar el sistema de información de una empresa de software atendiendo las siguientes premisas

- La empresa crea proyectos para otras empresas. De dichas empresas se almacena el CIF, nombre, dirección y teléfono así como un código interno de empresa.
- Los proyectos se inician en una determinada fecha y finalizan en otra. Además al planificarle se almacena la fecha prevista de finalización (que puede no coincidir con la finalización real)
- Los proyectos los realizan varios trabajadores, cada uno de ellos desempeña una determinada profesión en el proyecto (analista, jefe de proyecto, programador,...), dicha profesión tiene un código de profesión. En el mismo proyecto puede haber varios analistas, programadores,...
- Todos los trabajadores tienen un código de trabajador, un dni, un nombre y apellidos. Su profesión puede cambiar según el proyecto: en uno puede ser jefe y en otro un programador
- Se anota las horas que ha trabajado cada trabajador en cada proyecto.
- Puede haber varios proyectos que comiencen el mismo día.
- A todas las empresas les hemos realizado al menos un proyecto
- Todos los trabajadores han participado en algún proyecto
- En la base de datos, la profesión “administrador de diseño” no la ha desempeñado todavía ningún trabajador o trabajadora.

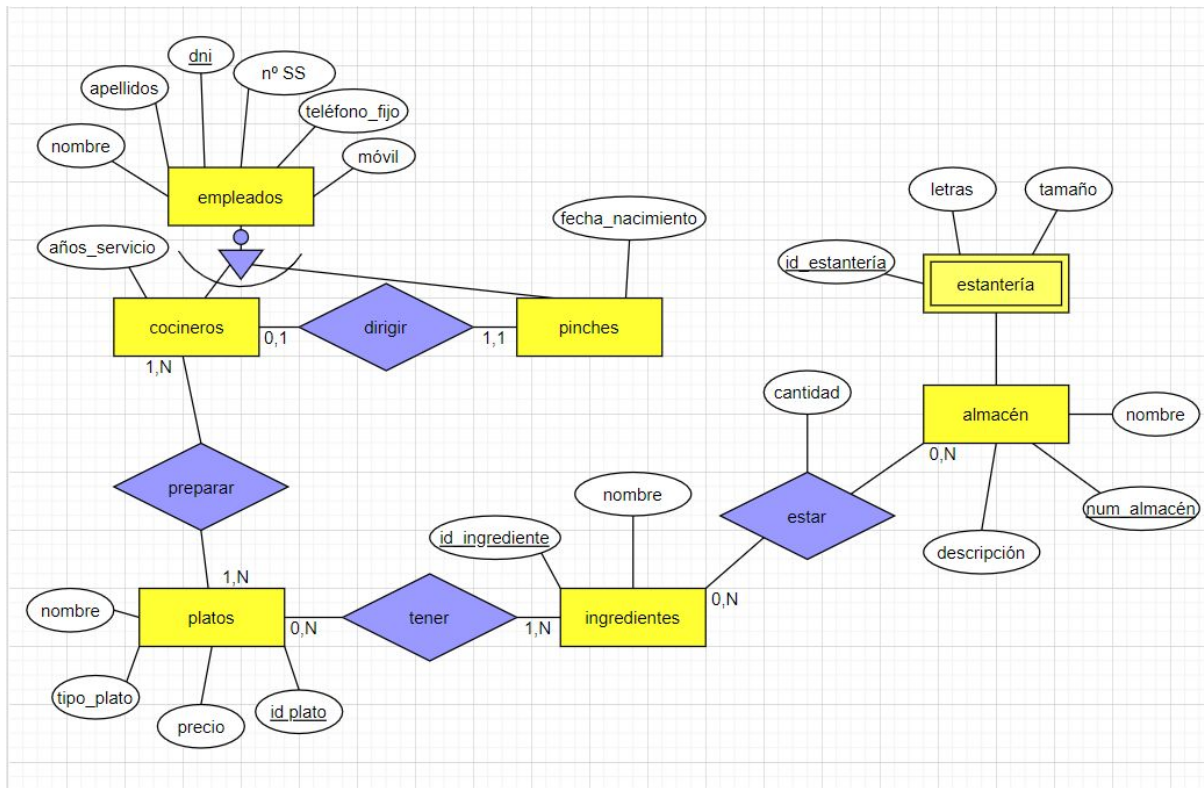


## Ejercicio 13. Empresa de comidas

Crear un diseño entidad/relación para una empresa de comidas. En la base de datos tienen que figurar:

- El nombre y apellidos de cada empleado, su dni y su número de SS además del teléfono fijo y el móvil
- Algunos empleados/as son cocineros/as. De los cocineros y cocineras anotamos (además de los datos propios de cada empleado) sus años de servicio en la empresa.
- Hay empleados/as que son pinches. De los y las pinches anotamos su fecha de nacimiento.
- La mayoría de trabajadores no son ni pinches ni cocineros/as
- En la base de datos figura cada plato (su nombre como “pollo a la carloteña”, “bacalo al pil-pil”,...), el precio del plato junto con los ingredientes que lleva. Anotamos también si cada plato es un entrante, un primer plato, segundo plato o postre
- De los ingredientes necesitamos la cantidad que necesitamos de él en cada plato y en qué almacén y estantería del mismo le tenemos.
- Cada almacén se tiene un nombre (despensa principal, cámara frigorífica A, cámara frigorífica B...), un número de almacén y una descripción del mismo.
- Cada estante en el almacén se identifica con dos letras y un tamaño en centímetros. Dos almacenes distintos pueden tener dos estantes con las mismas letras.
- Necesitamos también saber qué cocineros son capaces de preparar cada plato.
- Cada pinche está a cargo de un cocinero o cocinera.
- La cantidad de ingredientes en cada estantería de un almacén se actualiza en la base de datos al instante. Si cogemos dos ajos de un estante, figurará al instante que tenemos dos ajos menos en ese estante. Es necesario por lo tanto saber los ingredientes (cuáles y en qué número) que tenemos en cada estante.

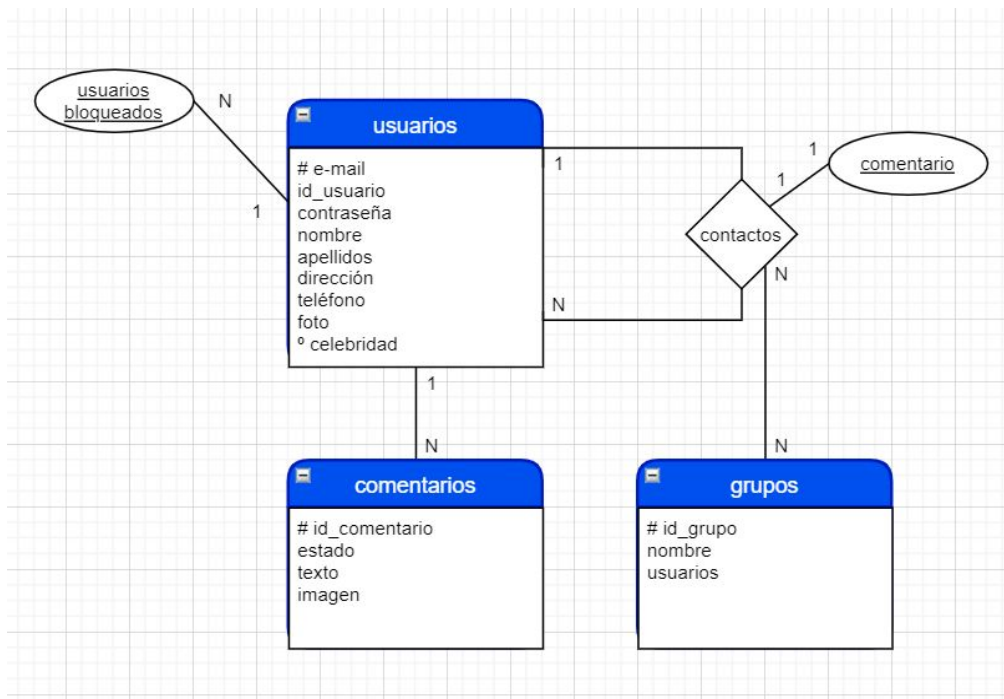




## Ejercicio 14. Red social

Crear un diseño entidad/relación que permita modelar un sistema que sirva para simular el funcionamiento de una red social, teniendo en cuenta lo siguiente:

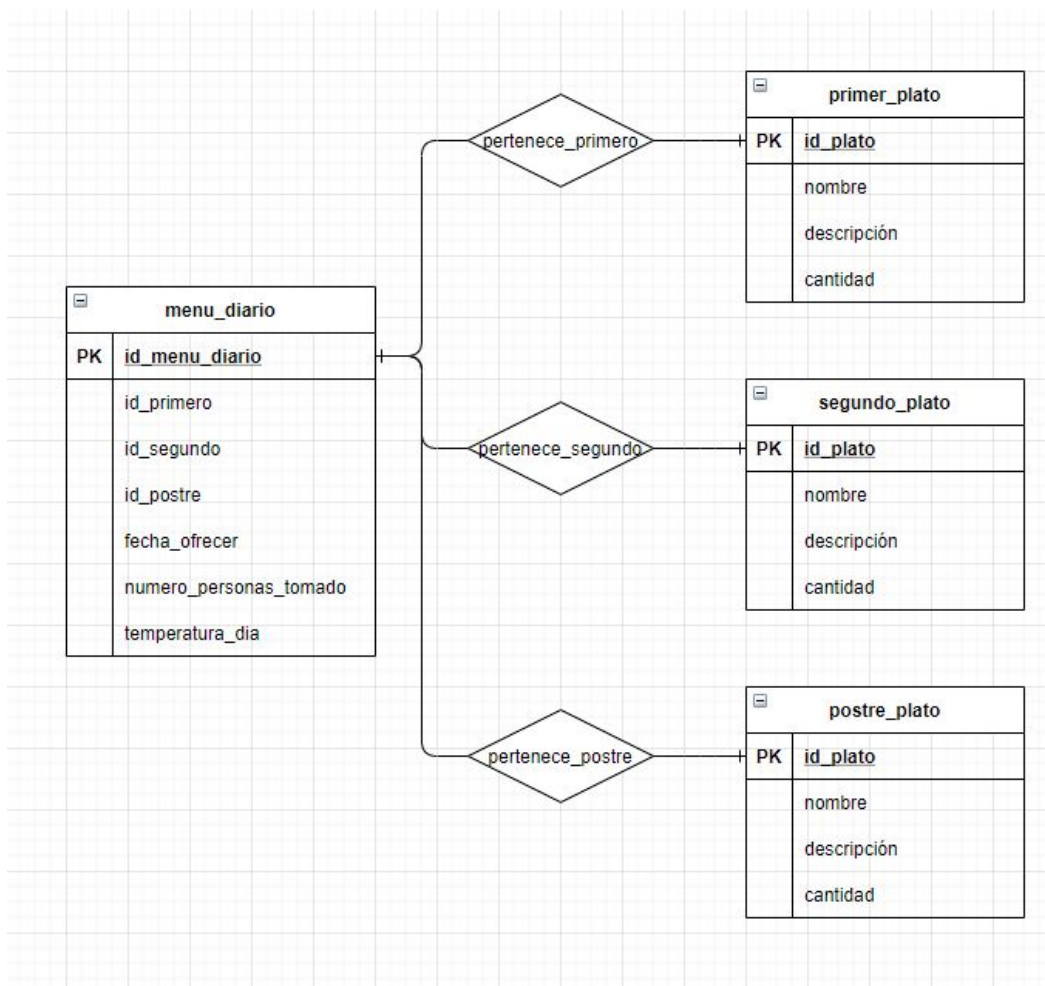
- Los usuarios de la red social se identifican con un identificador y una contraseña. Además se almacena de ellos:
  - Su nombre, apellidos, dirección, teléfono (puede tener varios teléfonos) e e-mail (el e-mail no tiene que poder coincidir con el de otro usuario) y una foto
  - Si los usuarios son celebridades, de ellos no aparecerá ni el email ni la dirección ni el teléfono.
- Los usuarios pueden tener una serie de contactos, que en realidad son otros usuarios. De cada contacto se puede almacenar un comentario que es personal y que sirve para describir al contacto.
- Los usuarios pueden organizar sus contactos en grupos de los cuales se almacena un nombre y deberemos saber los contactos que contiene. El mismo contacto puede formar parte de varios grupos.
- Además cada usuario puede tener una lista de usuarios bloqueados a fin de que no puedan contactar con él
- Los usuarios pueden publicar en la red comentarios, los cuales se puede hacer que los vea todo el mundo, que los vea uno o varios de los grupos de contactos del usuario o bien una lista concreta de usuarios. Los comentarios pueden incluir un texto y una imagen.



## Ejercicio 15. Menú diario

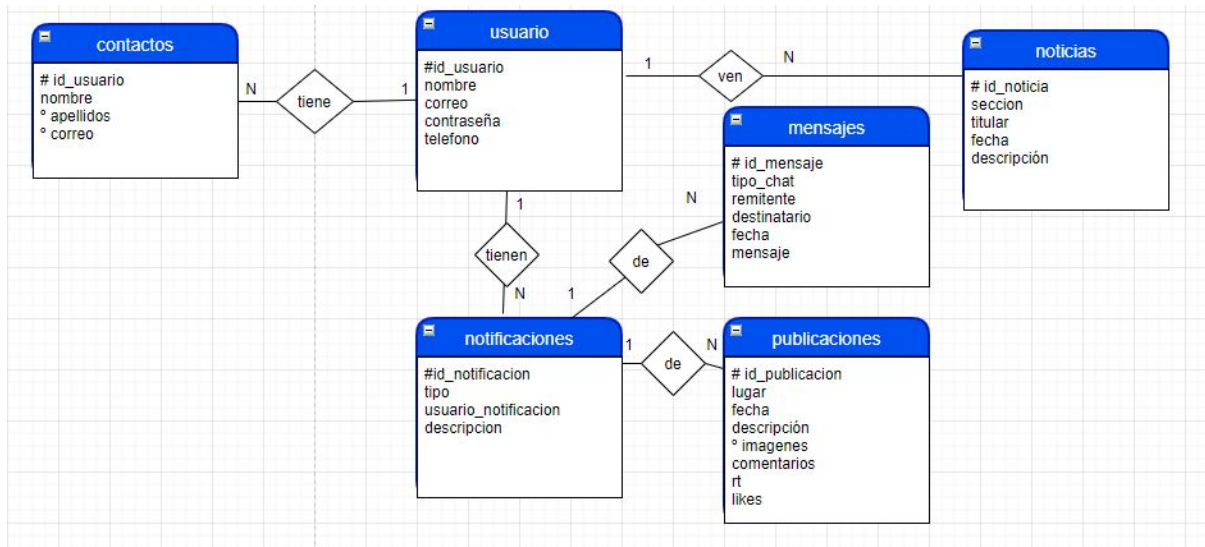
Crear un esquema Entidad/relación que represente un modelo para llevar los datos que maneja un restaurante de menús diarios. Teniendo en cuenta que:

- Sólo interesa llevar los datos de los menús diarios a la hora de la comida, nada más del restaurante
- Cada menú se compone de una serie de posibles platos. cada plato se puede repetir en diferentes días. Los platos pueden ser primer plato, segundo plato o postres.
- De cada plato se almacena el nombre (por ejemplo Arroz negro con setas) y una pequeña descripción.
- De los menús almacenamos la fecha en la que se ofrece el menú, el número de personas que han tomado menú ese día. Además almacenamos la cantidad de cada plato que se ha tomado ese día.
- Se almacena también la temperatura que hacía el día del menú para así poder analizar las temperaturas y los platos exitosos



## Ejercicio 16. Twitter

Crear un esquema Entidad/relación que represente un modelo para llevar los datos que maneja la red social Twitter: usuarios, mensajes,...



## Ejercicio 17. Horario escolar

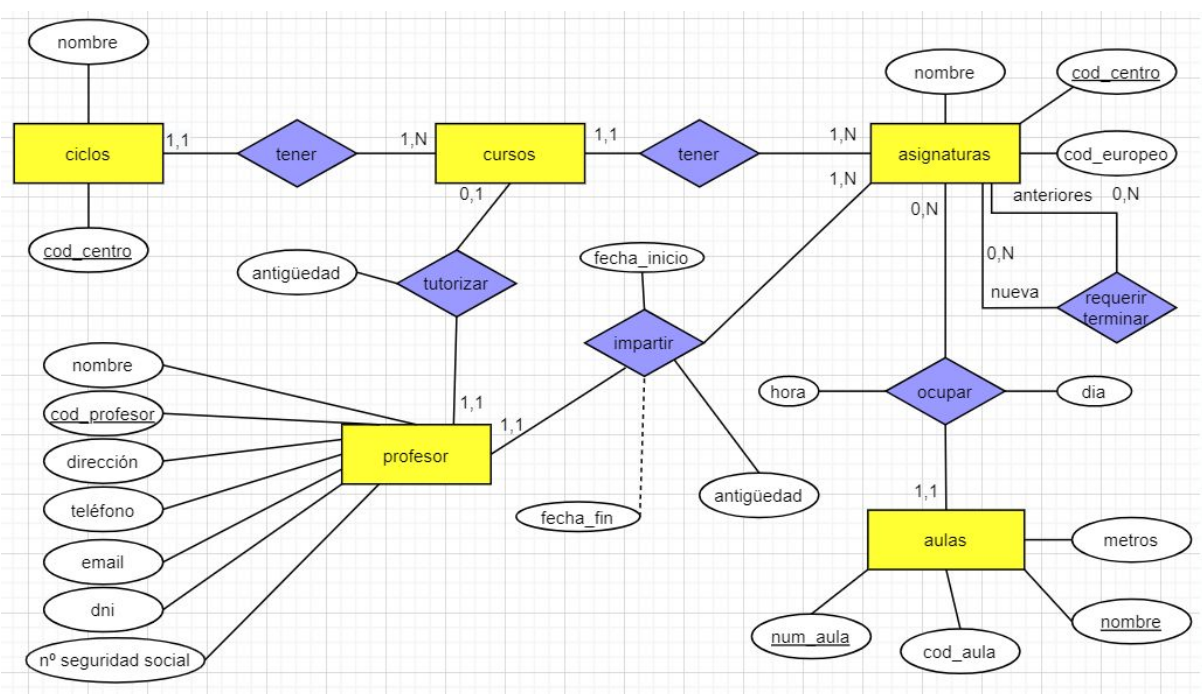
Crear un esquema Entidad/relación que represente el funcionamiento de un centro escolar de formación profesional, teniendo en cuenta que:

- Sólo interesa llevar el control de ocupación de las aulas en el horario escolar
- El horario es de seis horas diarias y en la base de datos simplemente se anota si es la primera, segunda,... y el día de la semana del que hablamos (por ejemplo miércoles a tercera hora)
- Las asignaturas tienen un nombre, un código interno del centro y un código europeo. La misma asignatura se puede impartir en dos ciclos distintos y en ese caso tendría el mismo código europeo y nombre, pero el código interno sería distinto. Hace falta saber en qué curso del ciclo se imparte la asignatura
- Los ciclos tienen un nombre, pueden ser de grado superior, de grado medio o de iniciación profesional; además tienen otro código interno en el centro.
- Las asignaturas en cada momento ocupan un aula, del que tenemos que almacenar un código de aula, un nombre (que no se repite), un número de aula (que tampoco se repite) y los metros que tiene. A una hora concreta de la semana, el aula puede estar vacía o bien ocuparse, pero sólo se puede ocupar por una asignatura
- Necesitamos saber y anotar en la base de datos si una asignatura requiere que antes se hayan aprobado otras, para poder matricularse en ella. Por ejemplo, **Ampliación de Matemáticas de 2o** a lo mejor requiere aprobar **Matemáticas de 1o**. Puede requerirse terminar más de una asignatura previamente para poder matricularse de una concreta.
- Se entiende que la asignatura sólo la puede impartir un profesor en todo el año, siempre será uno en todo momento el titular
- De los profesores se almacena su nombre, dirección, teléfono, email, DNI, no de Seguridad Social y un código interno de profesor así como los años que tiene de antigüedad impartiendo cada asignatura. Puede ser cada profesora o profesor, tutora de un curso y también se anota la antigüedad que tiene en esa tarea.

Complicamos el esquema anterior en este sentido

- Siendo más realistas, nos damos cuenta de que en un curso escolar, puede haber varios profesores responsables de una asignatura (por bajas, ceses, etc.); por lo que anotamos cuándo empezó a impartir dicho profesor la asignatura y cuando terminó (si no ha terminado, se dejaría vacío)
- Asegurar que podemos averiguar gracias al diseño, que si buscamos a un profesor un día concreto (por ejemplo el 13 de Mayo de 2012) a una hora concreta (sexta hora), podríamos saber en qué aula va a estar.





## Ejercicio 18. Inmuebles

Crear un diseño entidad/relación que permita modelar un sistema que sirva para gestionar una empresa que posee inmuebles. Para ello:

- Se almacenan los clientes usando su DNI, Teléfono fijo, Móvil, Nombre y Apellidos.
- Se almacenan los trabajadores y se almacenan los mismos datos. Ocurre además que un trabajador puede ser un cliente (porque puede alquilar o comprar mediante la inmobiliaria) a veces.
- A cada cliente y trabajador se le asigna un código personal
- Los clientes pueden comprar pisos, locales o garajes. En los tres casos se almacena un código de inmueble (único para cada inmueble), los metros que tienen, una descripción y su dirección.
- Los pisos tienen un código especial de piso que es distinto para cada piso.
- En los locales se indica el uso que puede tener y si tienen servicio o no.
- De los garajes se almacena el número de garaje (podría repetirse en distintos edificios) y la planta en que se encuentra (para el caso de garajes que están en varias plantas). Los garajes además pueden asociarse a un piso y así cuando se alquile el piso se incluirá el garaje.
- La empresa prevé que podría haber inmuebles que podrían no ser ni locales, ni garajes, ni pisos
- Los inmuebles se pueden comprar. Incluso varias veces. Se asigna un código de compra cada vez que se haga, la fecha y el valor de la compra. La compra puede tener varios titulares.
- Cada inmueble se puede alquilar y en ese caso se asigna un número de alquiler por cada inmueble. Ese número se puede repetir en distintos inmuebles (es decir puede haber alquiler no 18 para el inmueble 40 y el 35). Pero no se repite para el mismo inmueble.
- Al alquilar queremos saber el nombre del agente de la empresa que gestionó el alquiler así como a qué persona (solo una) estamos alquilando el inmueble.
- Cada pago de cada alquiler será almacenado en la base de datos, llevando el año, el mes y el valor del mismo.

