

Práctica 2: Ejercicios Modelo Entidad/Relación

(Solución)

En los siguientes ejercicios hay que construir la base de datos según el modelo Entidad/Relación. Tienes la solución de los 3 primeros, así que sólo hay que entregar a partir del 4º.

Primero: VENTA DE COCINAS (cocinas.dia)

Una empresa dedicada a comercializar cocinas desea aumentar su control sobre aquellos elementos que le afectan. Del resultado del análisis que realiza obtiene las siguientes informaciones:

Hay una serie de fabricantes de muebles de cocina. De cada fabricante se dispone de un nombre, una dirección y una relación de números de teléfono. Cada uno de ellos fabrica varios muebles de cocina. Un mueble de cocina tiene una determinada línea, un determinado color, unas dimensiones dadas (ancho x alto x largo), y puede tener una de las siguientes categorías excluyentes: mueble alto, mueble bajo, panel y encimera. De los muebles bajos interesa saber la altura sobre el suelo y de las encimeras interesa saber su tipo (mármol o aglomerado).

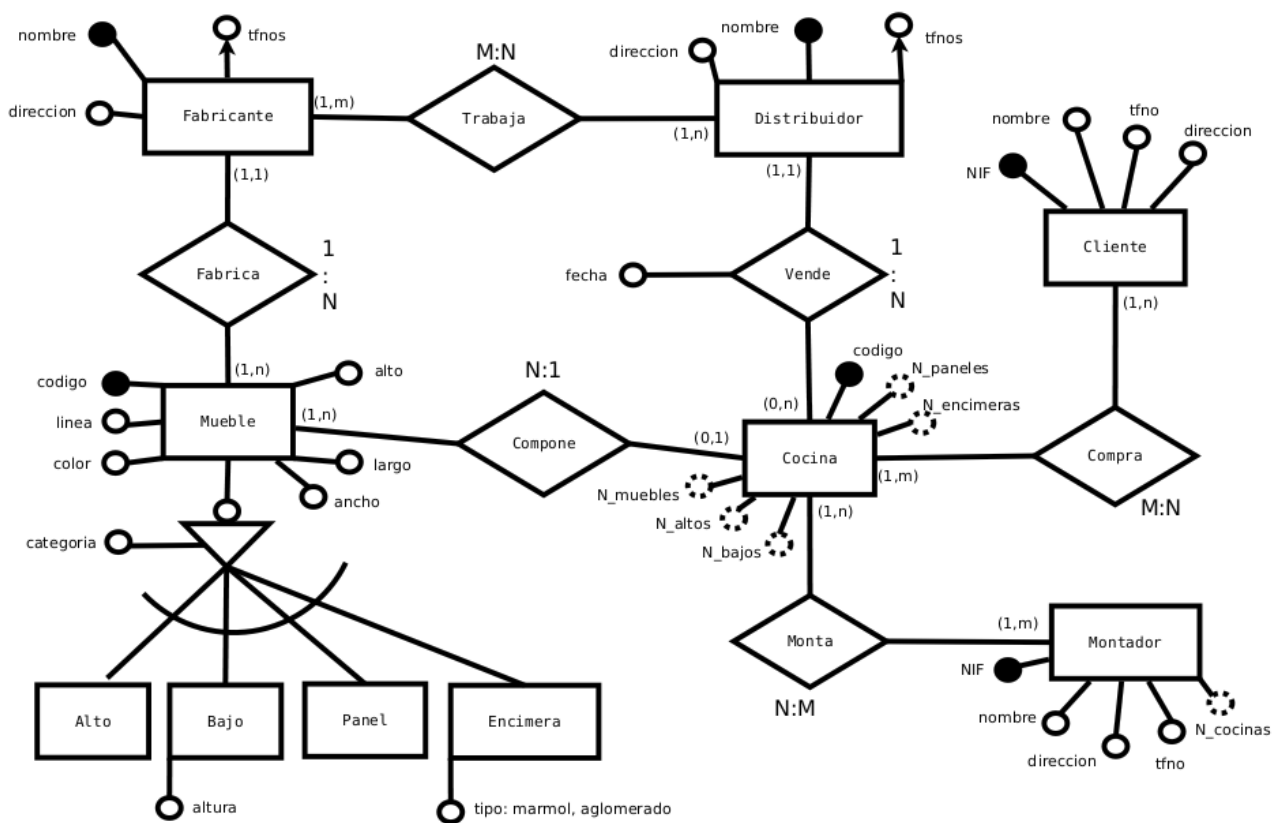
Cada fabricante puede trabajar con varios distribuidores y cada distribuidor trabaja al menos con un fabricante. De un distribuidor se dispone del nombre, dirección y una relación de números de teléfono.

Una cocina la componen una serie de muebles de cocina de distinto tipo, cada mueble de cocina sólo podrá formar parte de una única cocina. De una cocina nos interesa saber el número de muebles que la componen, así como cuántos de ellos hay de cada tipo.

Cada cocina la puede vender un único distribuidor en una determinada fecha de venta, aunque cada distribuidor puede vender varias cocinas.

Cada cocina la debe montar al menos un montador, y el mismo montador puede montar varias cocinas. De un montador nos interesa su NIF, nombre, dirección, único número de teléfono y el número de cocinas que ha montado.

Cada cocina pueden comprarla uno o varios clientes, y el mismo cliente puede comprar varias cocinas. De un cliente nos interesa su NIF, su nombre, dirección y único número de teléfono.



Segundo: CURSOS DE FORMACIÓN (cursos.dia)

El departamento de formación de una empresa desea construir una base de datos para planificar y gestionar la formación de sus empleados.

La empresa organiza cursos internos de formación de los que se desea conocer el código de curso, el nombre, una descripción, el número de horas de duración y el coste del curso.

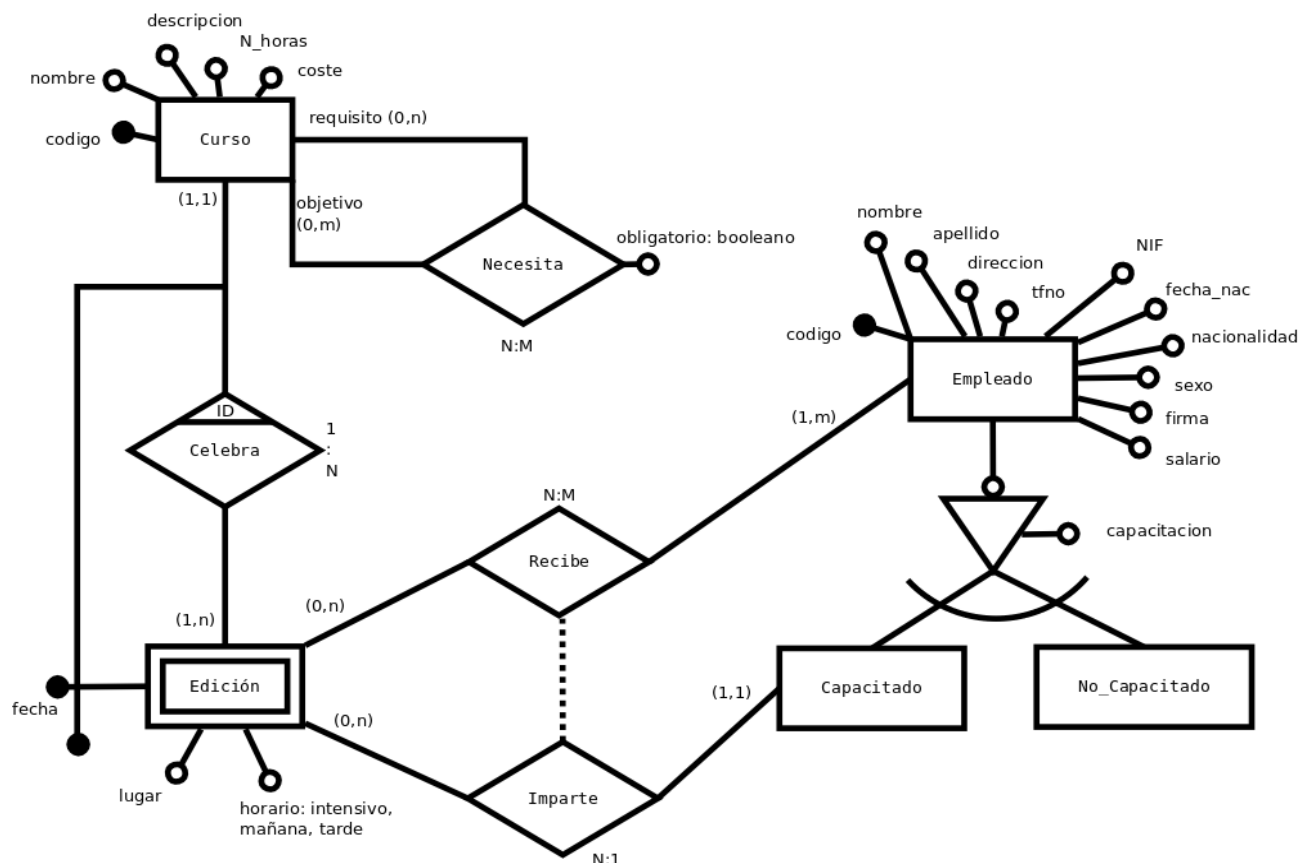
Un curso puede tener como prerequisite haber realizado otro(s) previamente, y, a su vez la realización de un curso puede ser prerequisite de otros. Un curso que es prerequisite de otro puede serlo de forma obligatoria o sólo recomendable.

Un mismo curso tiene diferentes ediciones, es decir, se imparte en diferentes lugares, fechas y con diferentes horarios (intensivos, de mañana o de tarde). En una misma fecha de inicio sólo puede impartirse una edición de un curso.

Los cursos se imparten por personal de la propia empresa.

De los empleados se desea almacenar su código de empleado, nombre y apellidos, dirección, teléfono, NIF (Número de Identificación Fiscal), fecha de nacimiento, nacionalidad, sexo, firma y salario, así como si está o no capacitado para impartir cursos.

Un mismo empleado puede ser docente en una edición de un curso y alumno en otra edición, pero nunca puede ser ambas cosas a la vez (en una misma edición de curso o lo imparte o lo recibe).



Tercero: ENTORNO DE EJECUCIÓN (entorno.dia)

Una empresa decide crear un único entorno de ejecución que controle la seguridad de acceso para todas sus aplicaciones informáticas. Para ello considera conveniente dividir sus aplicaciones en subsistemas funcionales especializados y establecer el control de acceso al nivel de estos subsistemas. Se desarrollará un motor de ejecución que, tomando como parámetros los contenidos de la BD, controlará la ejecución de los subsistemas y el acceso a los mismos. Este motor se hará cargo también de la navegación dentro de los subsistemas. Profundizando en este enfoque, se establecen los siguientes requisitos:

La unidad básica de acceso a los subsistemas es el denominado perfil de acceso. Un usuario tendrá acceso a todos los subsistemas a los que permiten acceder los distintos perfiles de que disfruta (al menos uno). Un perfil permite el acceso de al menos un subsistema y para cualquier subsistema habrá siempre un perfil que permita acceder al mismo.

De cada usuario se mantiene el DNI, nombre, teléfono y terminales en que trabaja.

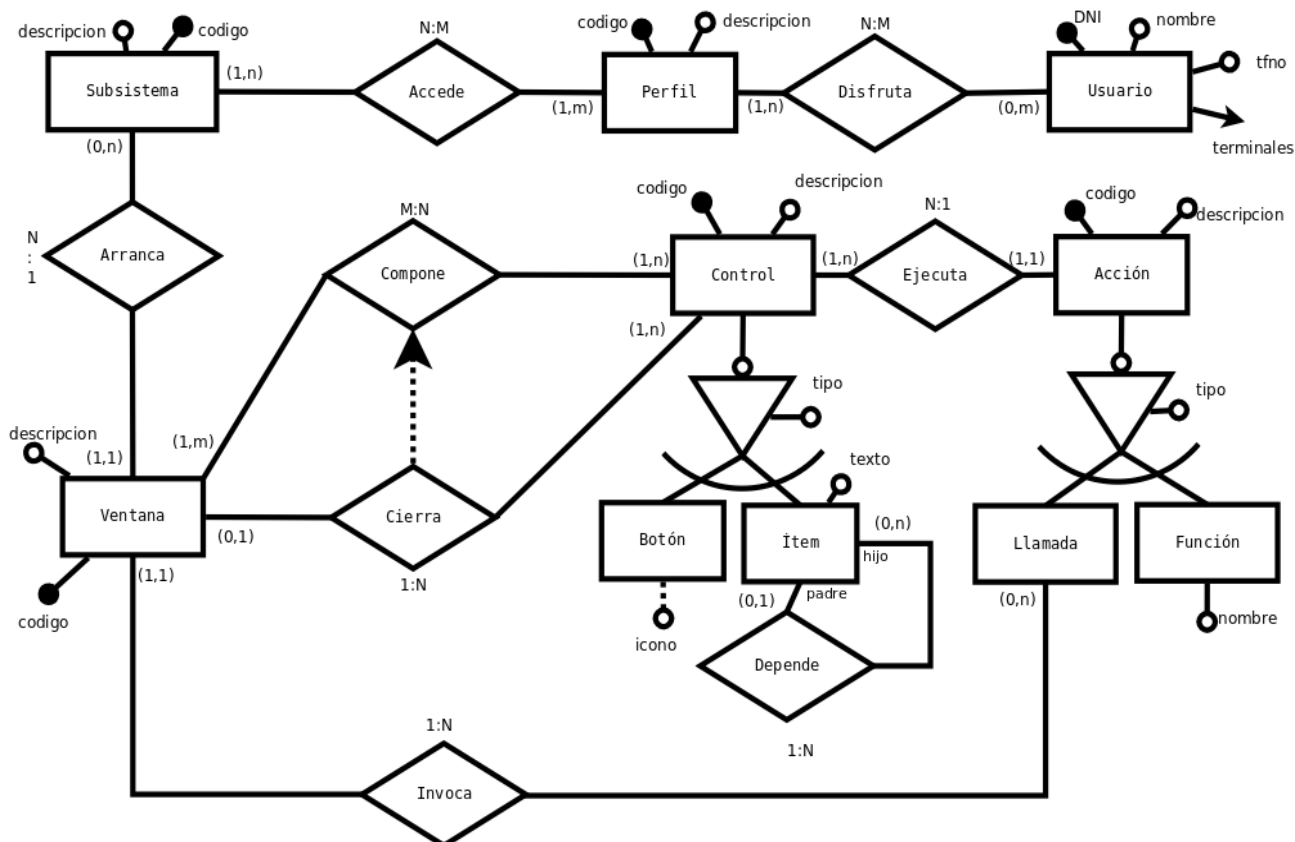
De los perfiles de acceso, lo mismo que de los subsistemas, se mantiene un código y una descripción. De los subsistemas, se mantiene, además, la ventana en la que arranca.

Las ventanas están compuestas por controles; toda ventana tendrá un control que permita cerrarla. Todo control ha de emplearse en alguna ventana y el mismo control puede ser empleado en distintas ventanas. De las ventanas y controles se mantiene también un código y una descripción.

Los controles pueden ser de dos tipos: botones o ítems de menú. Para soportar la estructura jerárquica de menús, de un ítem de menú pueden depender otros ítems, pero no puede darse la situación de que el mismo ítem dependa de varios ítems. En los ítems de menú se ha de mantener forzosamente el texto que se visualizará en pantalla. De los controles de tipo botón se mantiene el nombre del icono que opcionalmente se visualizará.

La activación de un control tiene como consecuencia la ejecución de una única acción (todo control ejecutará una acción al menos). Una acción requiere siempre un control que pueda ejecutarla. De las acciones se mantiene el código y la descripción.

Las acciones pueden ser de dos tipos, de función y de llamada. Las acciones de función ejecutan una función interna del propio entorno (de la que se ha de guardar el nombre). Las acciones de llamada invocan una única ventana.

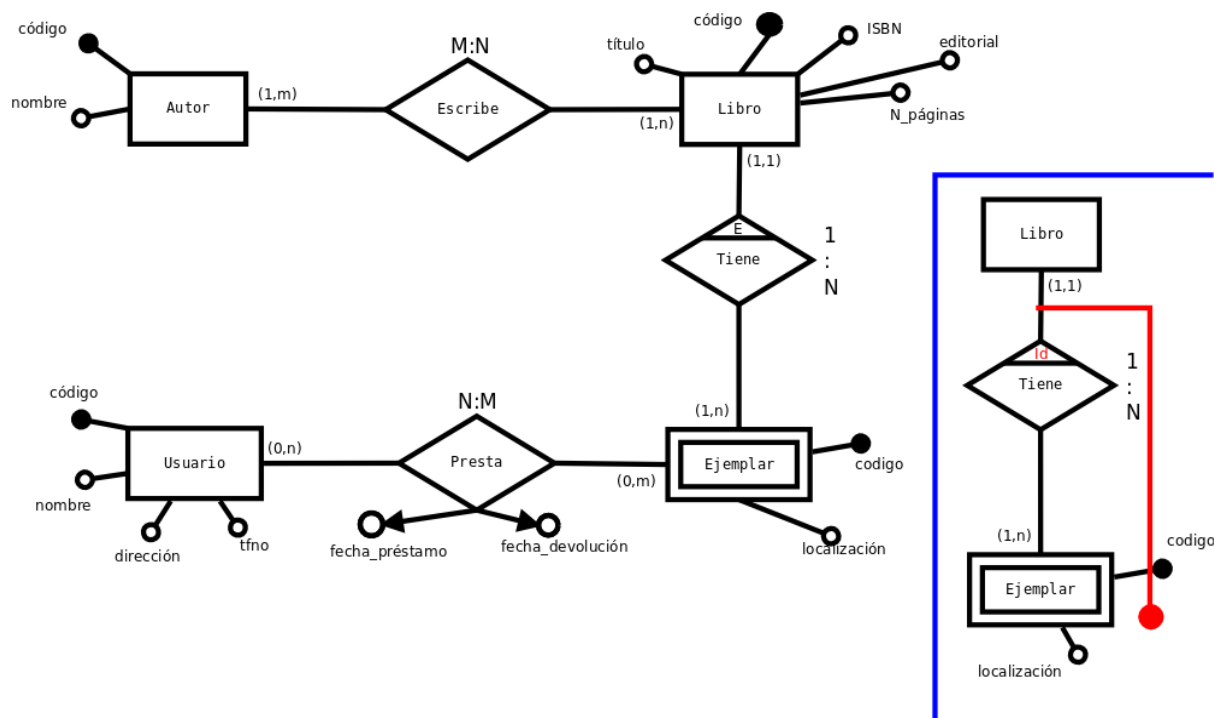


Cuarto: BIBLIOTECA (*biblioteca.dia*)

En la biblioteca del centro se manejan fichas de autores y libros. En la ficha de cada autor se tiene el código de autor y el nombre. De cada libro se guarda el código, título, ISBN, editorial y número de páginas. Un autor puede escribir varios libros, y un libro puede ser escrito por varios autores. De cada libro podemos disponer de varios ejemplares. Cada ejemplar tiene un código y una localización. Un libro tiene muchos ejemplares y un ejemplar pertenece sólo a un libro.

Los usuarios de la biblioteca del centro también disponen de ficha en la biblioteca y sacan ejemplares de ella. De cada usuario se guarda el código, nombre, dirección y teléfono.

Los ejemplares son prestados a los usuarios. Un usuario puede tomar prestados varios ejemplares, y un ejemplar puede ser prestado a varios usuarios. De cada préstamo interesa guardar la fecha de préstamo y la fecha de devolución.



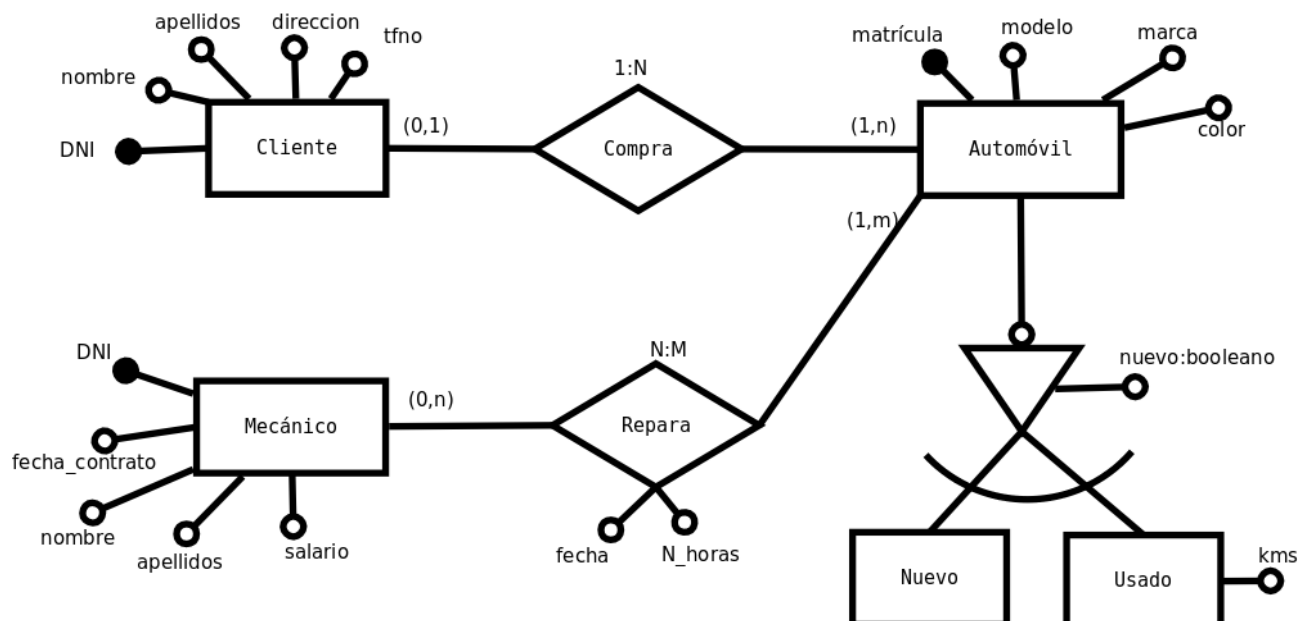
A la derecha hay una alternativa (las diferencias con el original están en rojo) en la que la relación TIENE es (además de relación de existencia) una relación de identificación. La elección depende en exclusiva del código del ejemplar: si se trata de un código único para cualquier ejemplar de cualquier libro de la biblioteca, será dependencia de existencia. Si el código del ejemplar puede repetirse entre ejemplares de libros distintos, será dependencia de identificación.

Quinto: CONCESIONARIO (*concesionario.dia*)

A un concesionario de coches llegan clientes para comprar automóviles. De cada coche interesa saber la matrícula, modelo, marca y color. Un cliente puede comprar varios coches en el concesionario. Cuando un cliente compra un coche, se le hace una ficha en el concesionario con la siguiente información: DNI, nombre, apellidos, dirección y teléfono.

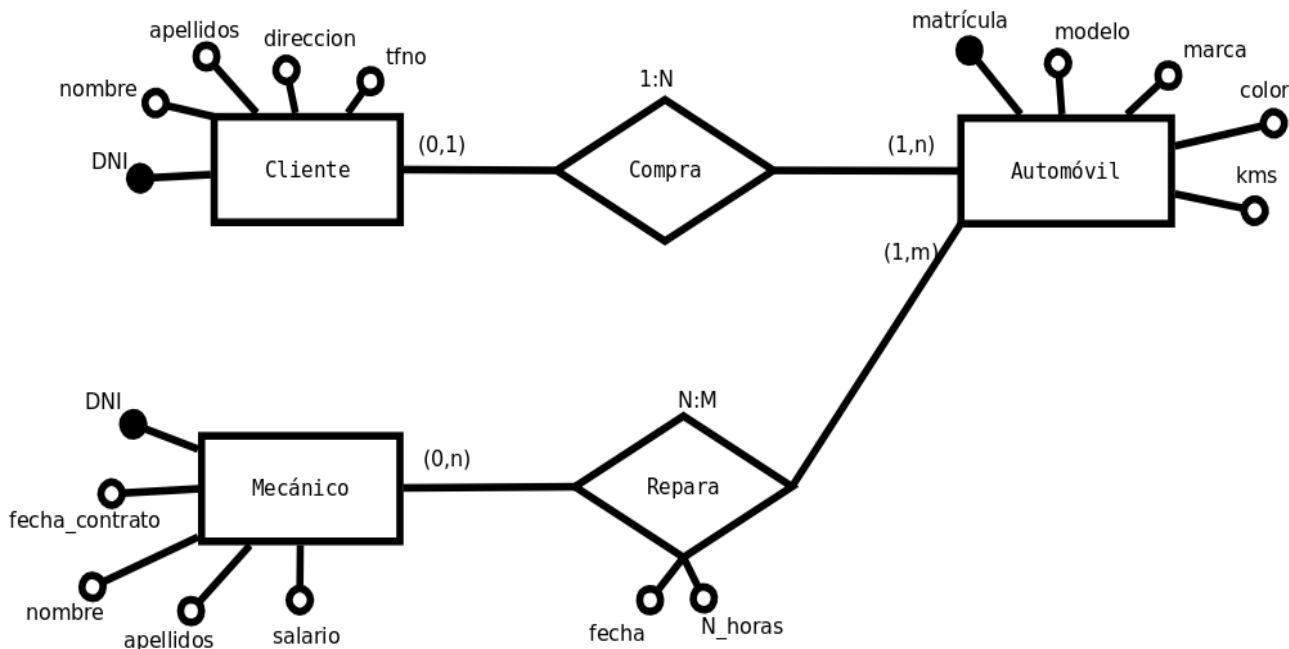
Los coches que el concesionario vende pueden ser nuevos o usados (de segunda mano). De los coches nuevos interesa saber el número de unidades que hay en el concesionario. De los coches viejos interesa el número de kilómetros que lleva recorridos.

El concesionario también dispone de un taller en el que los mecánicos reparan los coches que llevan los clientes. Un mecánico repara varios coches a lo largo del día, y un coche puede ser reparado por varios mecánicos. Los mecánicos tienen un DNI, nombre, apellidos, fecha de contratación y salario. Se desea guardar también la fecha en la que se repara cada vehículo y el número de horas que se ha tardado en arreglar cada automóvil.



La frase “de los coches nuevos interesa saber el número de unidades” hace referencia al número de ocurrencias de la entidad NUEVO. Se trata de un atributo derivado que se puede calcular cuando sea necesario mostrarlo (ver SELECT y count() en SQL). En ningún caso se trata de un atributo de la entidad NUEVO puesto que no es una característica de un vehículo.

Una vez que sabemos que no hay atributos distintos para vehículos nuevos o usados, una solución más sencilla (y más acertada) sería eliminar la jerarquía e incluir el atributo KMS en la entidad AUTOMÓVIL. Sabríamos los automóviles que son nuevos porque el número de kms. es cero.



Sexto: LIGA (liga.dia)

La liga de fútbol profesional, presidida por Don Ángel María Villar, ha decidido informatizar sus instalaciones creando una base de datos para guardar la información de los partidos que se juegan en la liga.

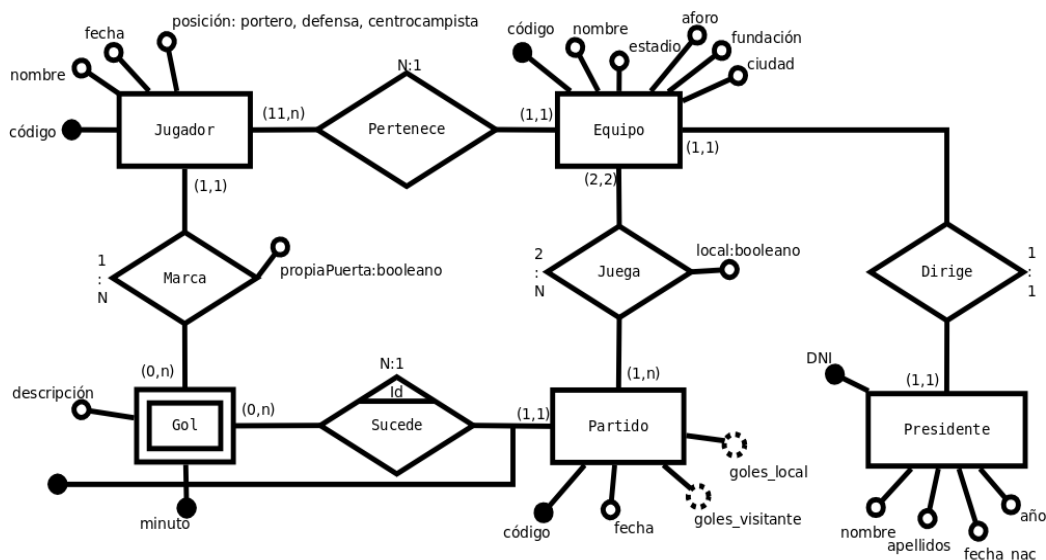
Se desea guardar en primer lugar los datos de los jugadores. De cada jugador se quiere guardar el nombre, fecha de nacimiento y posición en la que juega (portero, defensa, centrocampista...). Cada jugador tiene un código de jugador que lo identifica de manera única.

De cada uno de los equipos de la liga es necesario registrar el nombre del equipo, nombre del estadio en el que juega, el aforo que tiene, el año de fundación del equipo y la ciudad de la que es el equipo. Cada equipo también tiene un código que lo identifica de manera única. Un jugador sólo puede pertenecer a un único equipo.

De cada partido que los equipos de la liga juegan hay que registrar la fecha en la que se juega el partido, los goles que ha metido el equipo de casa y los goles que ha metido el equipo de fuera. Cada partido tendrá un código numérico para identificar el partido.

También se quiere llevar un recuento de los goles que hay en cada partido. Se quiere almacenar el minuto en el que se realiza el gol y la descripción del gol. Un partido tiene varios goles y un jugador puede meter varios goles en un partido.

Por último se quiere almacenar, en la base de datos, los datos de los presidentes de los equipos de fútbol (DNI, nombre, apellidos, fecha de nacimiento, equipo del que es presidente y año en el que fue elegido presidente). Un equipo de fútbol tan sólo puede tener un presidente, y una persona sólo puede ser presidente de un equipo de la liga.



El año de elección de un presidente podría ser una característica de la relación DIRIGE y no del presidente (porque un presidente puede iniciar la dirección en varios periodos distintos). Sin embargo no contemplamos esta posibilidad porque el enunciado dice explícitamente “un equipo tan solo puede tener un presidente” lo que implica que la base de datos no guarda histórico de los presidentes, sino únicamente el presidente en activo.

Observamos que se forma un ciclo con las relaciones PERTENECE, JUEGA, SUCEDE y MARCA. Así que tenemos que analizar si alguna de ellas es redundante. Observamos que no podemos quitar ninguna de ellas sin perder semántica, por tanto, el ciclo se queda.

Observamos que nos pide anotar (para cada partido) los goles que marca el equipo de casa y los que marca el equipo visitante. Sin embargo ya existe una entidad GOL y añadir los atributos goles_local y goles_visitante a la entidad PARTIDO sería redundante. Por eso los marcamos como atributos derivados: los dibujamos para dejar constancia de que atendemos esa solicitud expresa, pero no son atributos reales que se guardan en la base de datos, sino que se calculan a partir de las relaciones PARTIDO sucede GOL marca JUGADOR pertenece EQUIPO juega PARTIDO. Al seguir este camino se nos hace necesario incluir los atributos propiaPuerta en la relación MARCA y local en la relación JUEGA para poder determinar en cada una de las ocurrencias de GOL, si es un gol en casa o un gol fuera. Esos atributos, por otro lado, no son demasiado extraños, es más, aportan información realmente importante.

Séptimo: ACADEMIA (academia.dia)

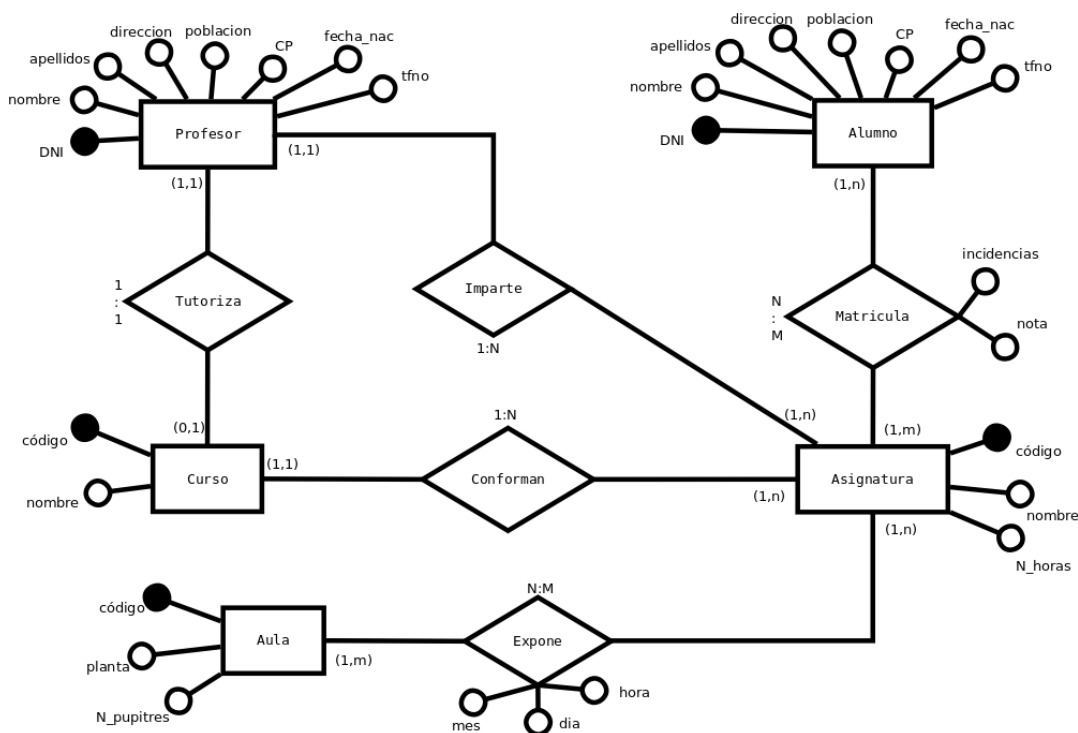
Se desea informatizar la gestión de un centro de enseñanza para llevar el control de los alumnos matriculados y los profesores que imparten clases en ese centro. De cada profesor y cada alumno se desea recoger el nombre, apellidos, dirección, población, DNI, fecha de nacimiento, código postal y teléfono.

Los alumnos se matriculan en una o más asignaturas, y de ellas se desea almacenar el código de asignatura, nombre y número de horas que se imparten a la semana. Un profesor del centro puede impartir varias asignaturas, pero una asignatura sólo es impartida por un único profesor. De cada una de las asignaturas se desea almacenar también la nota que saca el alumno y las incidencias que puedan darse con él.

Además, se desea llevar un control de los cursos que se imparten en el centro de enseñanza. De cada curso se guardará el código y el nombre. En un curso se imparten varias asignaturas, y una asignatura sólo puede ser impartida en un único curso.

Las asignaturas se imparten en diferentes aulas del centro. De cada aula se quiere almacenar el código, piso del centro en el que se encuentra y número de pupitres de que dispone. Una asignatura se puede dar en diferentes aulas, y en un aula se pueden impartir varias asignaturas. Se desea llevar un registro de las asignaturas que se imparten en cada aula. Para ello se anotará el mes, día y hora en el que se imparten cada una de las asignaturas en las distintas aulas.

La dirección del centro también designa a varios profesores como tutores en cada uno de los cursos. Un profesor es tutor tan sólo de un curso. Un curso tiene un único tutor. Se habrá de tener en cuenta que puede que haya profesores que no sean tutores de ningún curso.



Octavo: EMPRESA (empresa2.dia)

La empresa está organizada en una serie de departamentos. Cada departamento tiene un código, nombre y presupuesto anual. Cada departamento está ubicado en un centro de trabajo. La información que se desea guardar del centro de trabajo es el código de centro, nombre, población y dirección del centro.

La empresa tiene una serie de empleados. Cada empleado tiene un teléfono, fecha de alta en la empresa, NIF y nombre. De cada empleado también interesa saber el número de hijos que tiene y el salario de cada empleado.

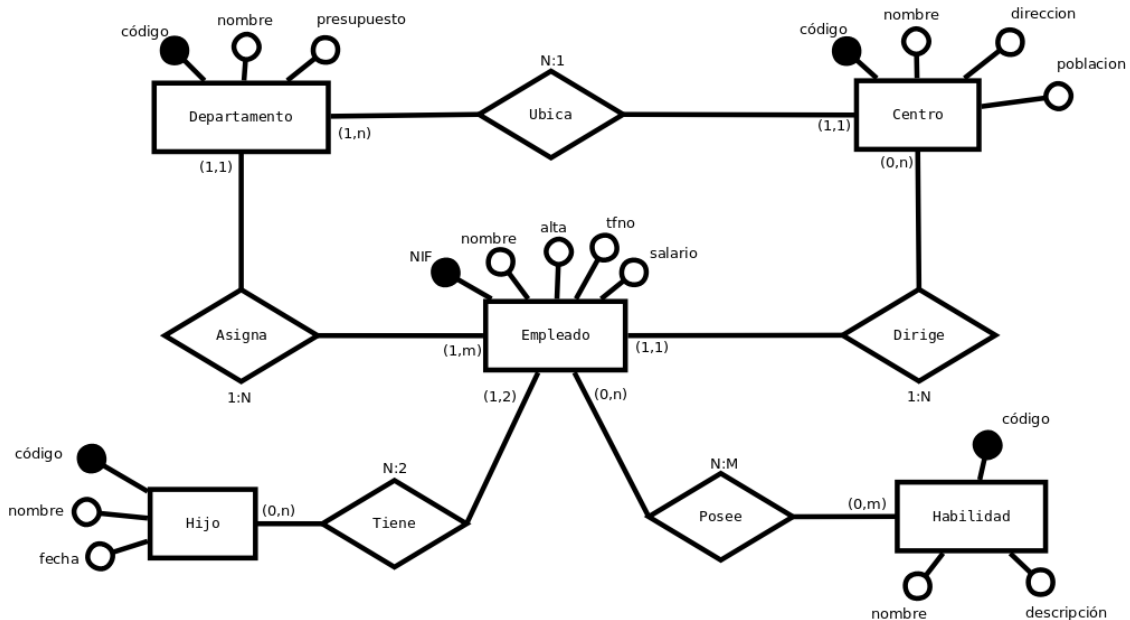
A esta empresa también le interesa tener guardada información sobre los hijos de los empleados. Cada hijo de un empleado tendrá un código, nombre y fecha de nacimiento.

Se desea mantener también información sobre las habilidades de los empleados (por ejemplo, mercadotecnia, trato con el cliente, fresador, operador de telefonía, etc...). Cada habilidad tendrá una descripción y un código.

Sobre este supuesto diseñar el modelo E/R y el modelo relacional teniendo en cuenta los siguientes aspectos.

- Un empleado está asignado a un único departamento. Un departamento estará compuesto por uno o más empleados.
- Cada departamento se ubica en un único centro de trabajo. Estos se componen de uno o más departamentos.
- Un empleado puede tener varios hijos.

- Un empleado puede tener varias habilidades, y una misma habilidad puede ser poseída por empleados diferentes.
- Un centro de trabajo es dirigido por un empleado. Un mismo empleado puede dirigir centros de trabajo distintos.



N_hijos de empleado es un atributo derivado (número de ocurrencias de la relación TIENE).

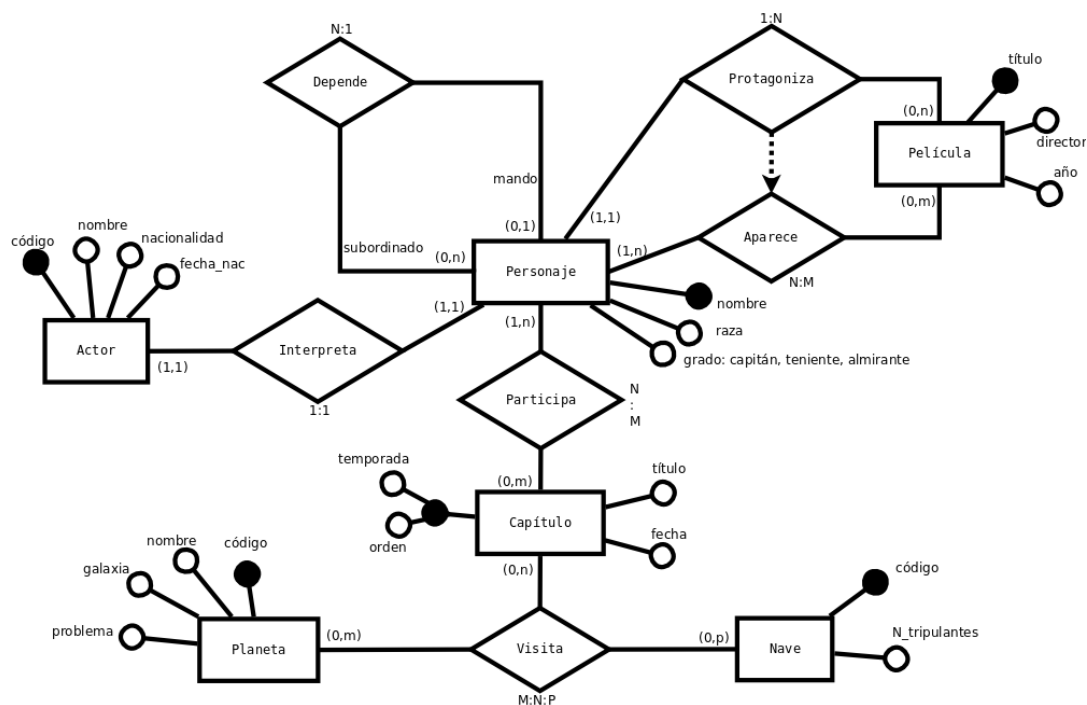
En este ejercicio se habría admitido HIJO como una entidad débil de EMPLEADO, con participación (1,1) en la parte del empleado. Sin embargo, en esta solución hemos sido más exhaustivos y contemplamos la posibilidad que trabajen en nuestra empresa ambos progenitores del hijo, la participación es (1,2) y deja de existir la dependencia de existencia y por tanto, la entidad débil.

Noveno: Startrek I (startrek1.dia)

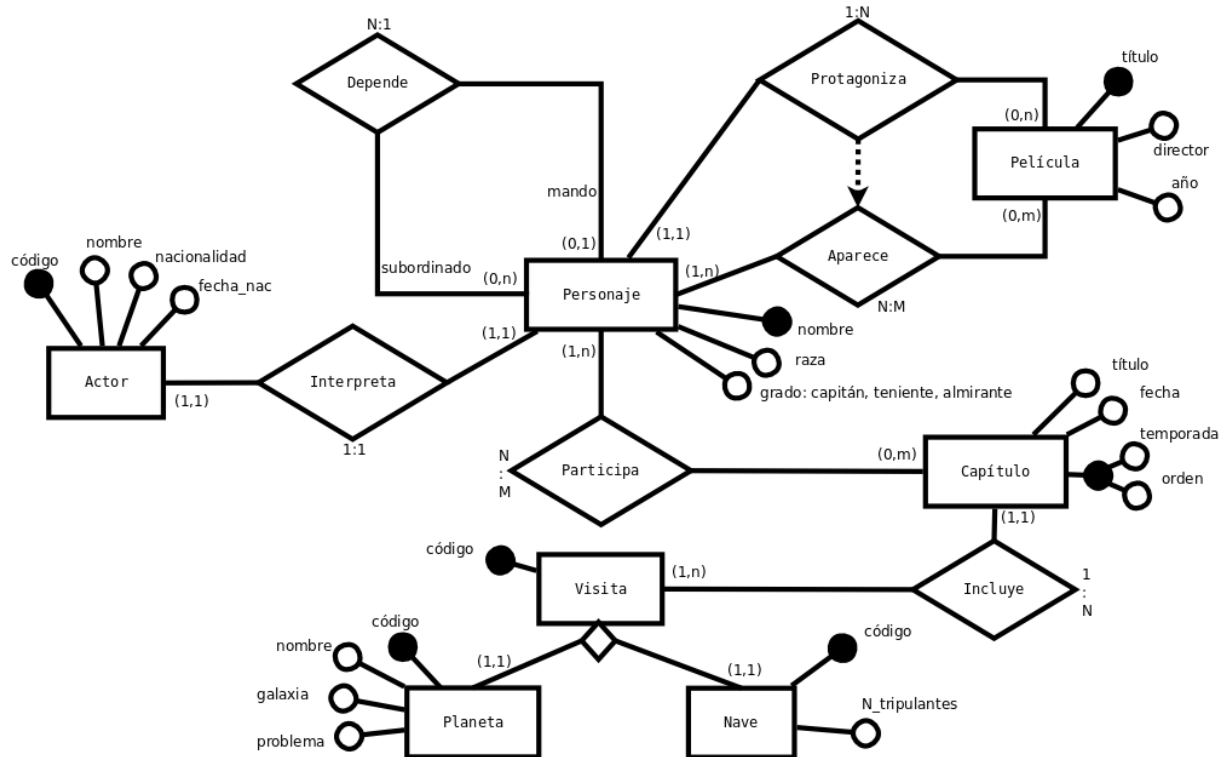
Un club de fans de la famosa película Startrek, ha decidido crear una página web donde se pueda consultar información referente a todas las películas y capítulos de la saga. El dominio startrekfans.com se redigirá a un servidor web que consulte una base de datos con la siguiente información:

- **Actores:** Es necesario conocer el Nombre completo del actor, el personaje que interpreta, la fecha de nacimiento y su nacionalidad.
- **Personajes:** De los personajes es necesario saber el nombre, su raza y graduación militar que desempeña (capitán, teniente, almirante, etc.). Es importante conocer el actor que interpreta el personaje, teniendo en cuenta que, un personaje sólo puede ser interpretado por un actor, y un actor sólo puede interpretar un personaje. Además, será necesario conocer el personaje del que depende directamente en graduación militar.
- **Capítulos:** Hay que almacenar todos los capítulos, indicando a qué temporada pertenece cada capítulo, el título, el orden en el que fue rodado, fecha de su primera emisión en televisión y los personajes que participaron en cada capítulo.

- Películas: Se debe almacenar también, todas las películas que se proyectaron en cines, cada una con su año de lanzamiento, título y director. También hay que guardar los personajes que aparecen en cada película y cuál de ellos fue el protagonista.
- Planetas: En cada capítulo, se visitan 1 o varios planetas, hay que almacenar el código del planeta, su nombre, galaxia a la que pertenece, y el problema que se resolvió con esa visita y la nave con la que se viajó al planeta. Para la descripción del problema será suficiente con un campo de texto de 255 caracteres. De la nave se almacenará el código y número de tripulantes.

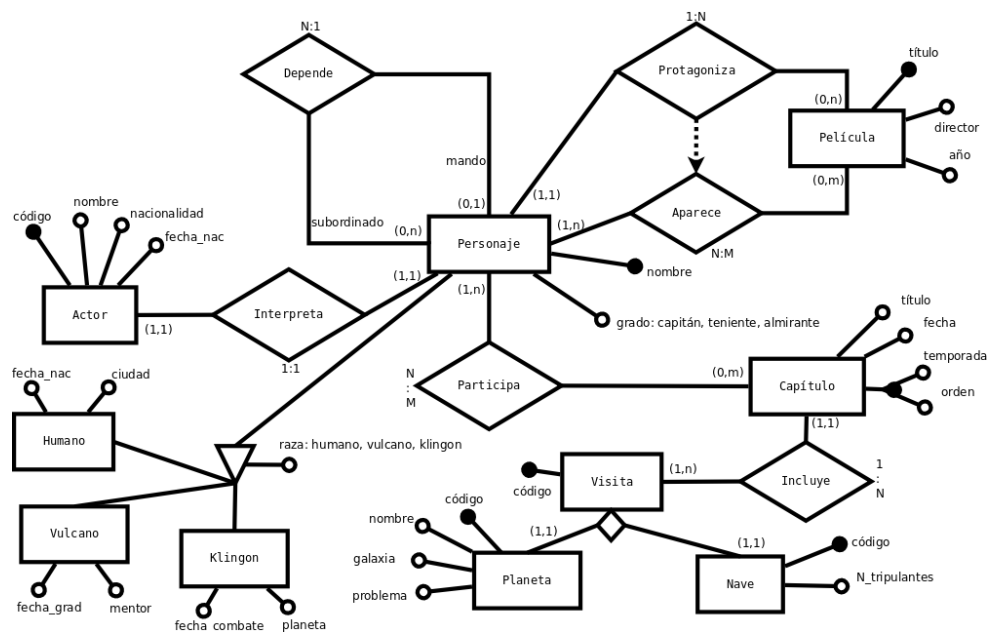


En la solución aportada, las cardinalidades de la relación ternaria VISITA no permiten incluir “en cada capítulo se visitan uno o varios planetas”. Para poder reflejar esto vamos a crear una entidad agregada VISITA compuesta de PLANETA y NAVE. A través de los componentes sabemos el planeta y la nave involucrados en cada visita y, además podemos concretar que en cada capítulo se harán una o más visitas:



Décimo: Starrek II (startrek2.dia)

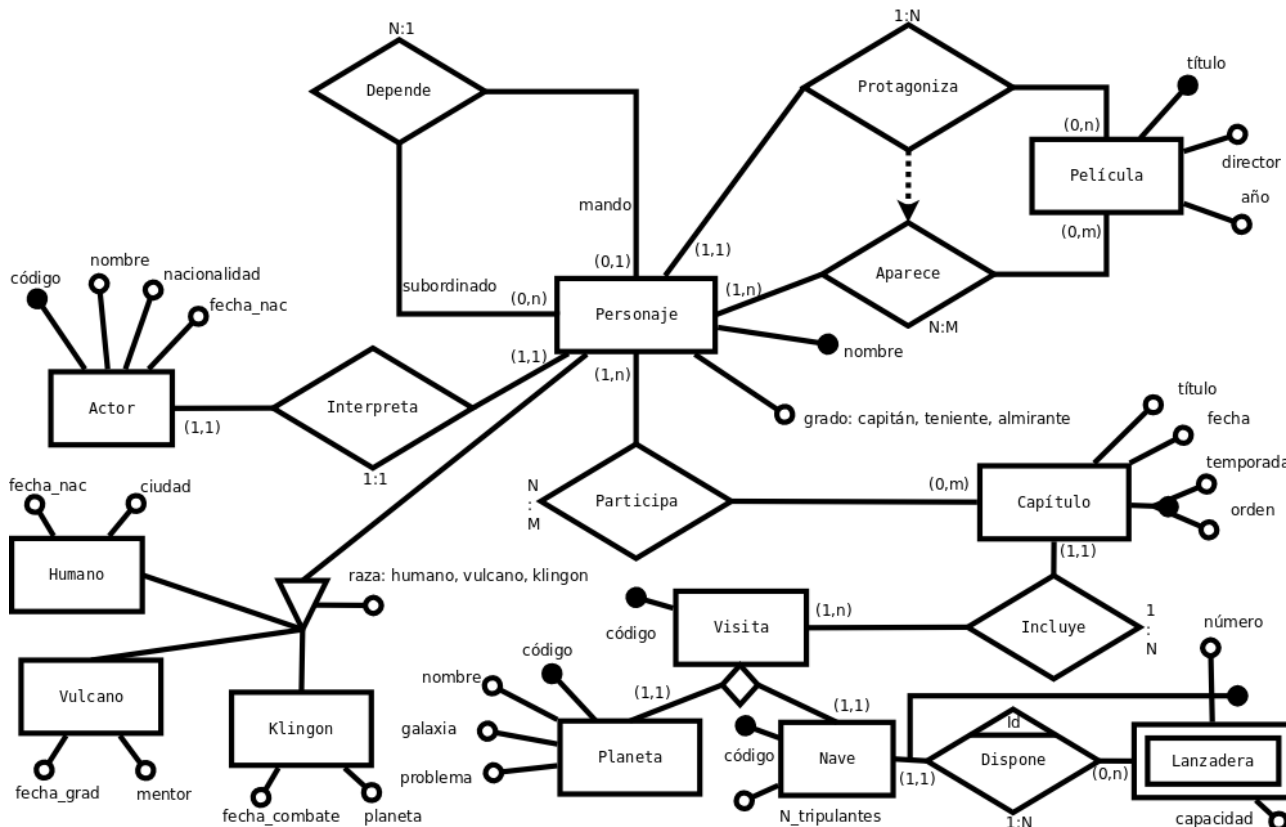
El club de fans de startrek ha pensado ampliar los requisitos de la página web para hacer una segunda versión. Esta segunda versión consiste en incluir información extra para los personajes. De esta manera, si el personaje es un humano, se indicará su fecha de nacimiento y ciudad terráquea donde nació. Si el personaje es de la raza Vulcano, se almacenará el nombre del mentor y la fecha de graduación, y si es de raza Klingon, se guardará su planeta natal y la fecha de su último combate.



Décimoprimer: Startrek III (startrek3.dia)

El club de fans de startrek quiere la tercera versión de la base de datos de la siguiente forma:

En cada capítulo, la nave que viaja a un planeta, puede disponer de naves pequeñas llamadas lanzaderas con las que bajan a la superficie del planeta. La existencia de la lanzadera, sólo tiene sentido si existe la nave a la que pertenece. Se identificará cada lanzadera mediante un número entero y el código de la nave. Es necesario conocer la capacidad de personas de la lanzadera.



Último:

Una vez terminados los ejercicios, crea una carpeta llamada **apellido1_apellido2**, copia a esa carpeta todos los ficheros generados y los ficheros necesarios si los hubiera (imágenes, ...), comprime la carpeta en un **ZIP** llamado **apellido1_apellido2.zip** y súbelo con el enlace correspondiente de la plataforma **Moodle**.

No debes añadir nada más (no uses subcarpetas, no pongas más datos en los nombres de ficheros o carpetas). **EL INCUMPLIMIENTO DE ESTAS INSTRUCCIONES PUEDE TENER COMO CONSECUENCIA LA NO CORRECCIÓN DE LA PRÁCTICA.**