

# Bases de datos - Práctica 1

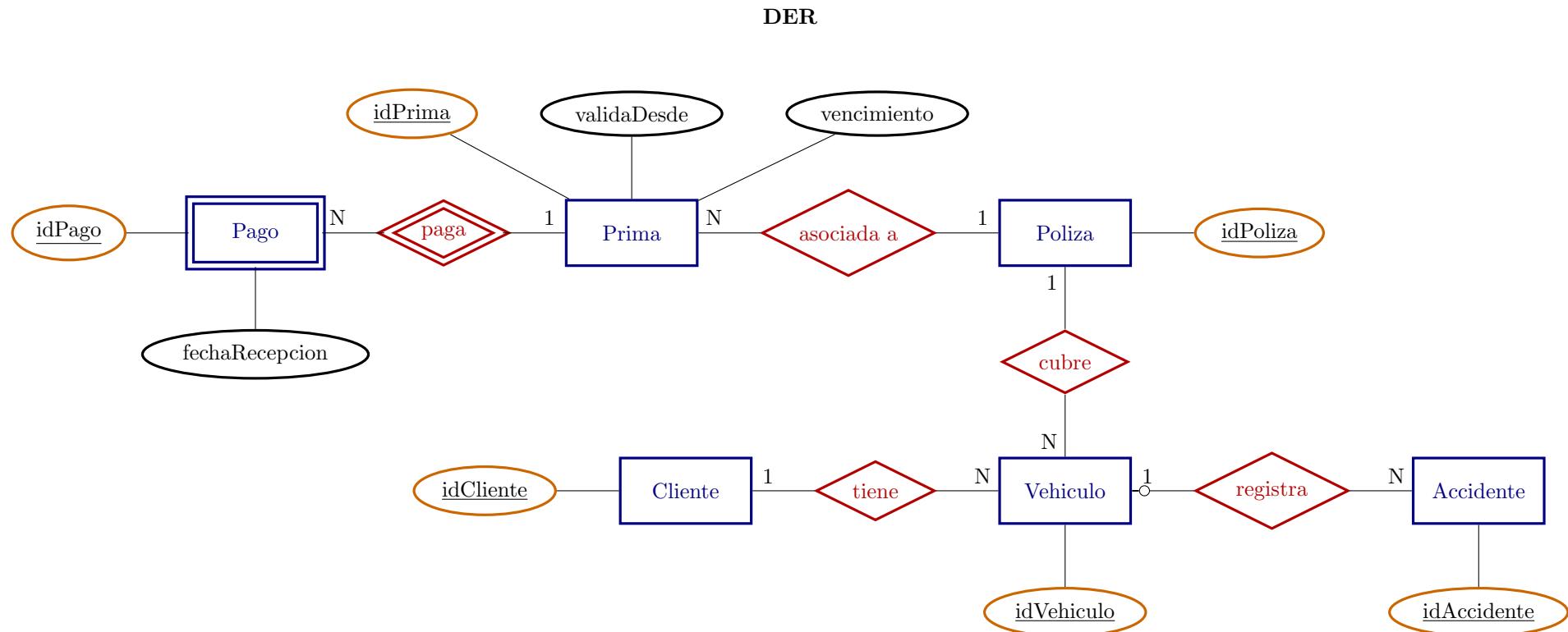
19 de agosto de 2019

## Índice

<b>1. Ejercicios introductorios</b>	<b>2</b>
1.1. Ejercicio 1.1 . . . . .	2
1.2. Ejercicio 1.2 . . . . .	3
1.3. Ejercicio 1.3 . . . . .	4
1.4. Ejercicio 1.4 . . . . .	5
1.5. Ejercicio 1.5 . . . . .	7
1.6. Ejercicio 1.6 . . . . .	9
1.7. <b>Ejercicio 1.7 - Discutir inciso e</b> . . . . .	11
<b>2. Ejercicios Avanzados</b>	<b>14</b>
2.1. Ejercicio 2.1 . . . . .	14
2.2. <b>Ejercicio 2.2 - Revisar</b> . . . . .	15
2.3. Ejercicio 2.3 . . . . .	16
2.4. <b>Ejercicio 2.4 - Revisar</b> . . . . .	17
2.5. Ejercicio 2.5 . . . . .	19

## 1. Ejercicios introductorios

### Ejercicio 1.1



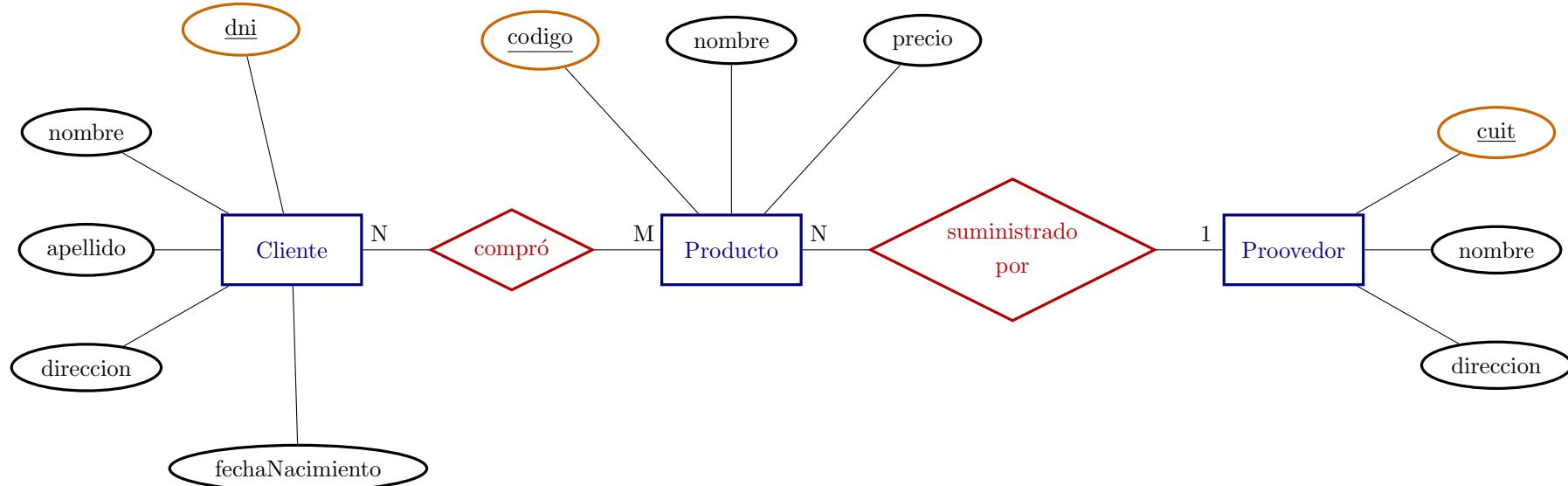
**MER**

**Cliente** (idCliente)  
**Vehiculo** (idVehiculo, idCliente, idPoliza)  
**Accidente** (idAccidente, idVehiculo)

**Poliza** (idPoliza)  
**Prima** (idPrima, validaDesde, vencimiento, idPoliza)  
**Pago** (idPago, idPrima, fechaRepcion, cantidad)

## Ejercicio 1.2

DER

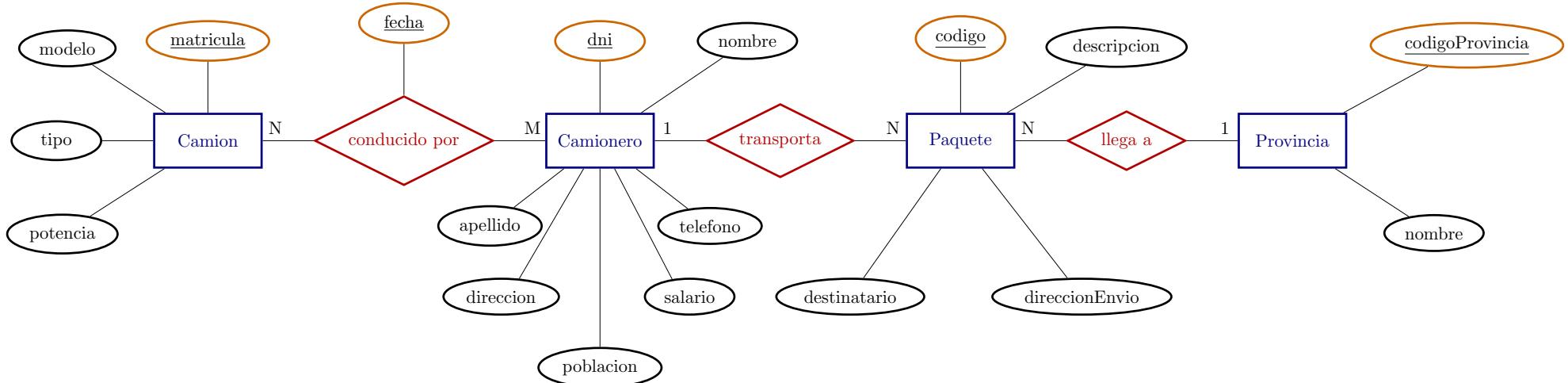


MER

Cliente (dni, nombre, apellido, direccion, nacimiento)  
Producto (codigo, nombre, precio, cuit)  
Proovedor (cuit, nombre, direccion)  
compro (dni, codigo)

## Ejercicio 1.3

DER

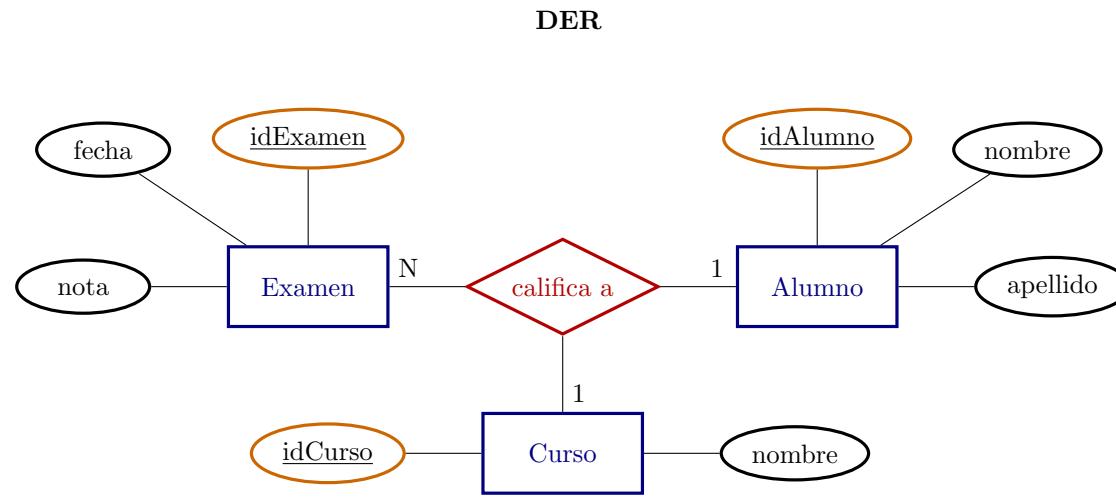


MER

Camion (matricula, modelo, tipo, potencia)  
 Camionero (dni, nombre, apellido, telefono, direccionEnvio, salario, poblacion)  
 Paquete (codigo, descripcion, destinatario, direccionDestinatario, dni, codigoProvincia)  
 Provincia (codigoProvincia, nombre)  
 compro(matricula, dni, fecha)

## Ejercicio 1.4

a)



Examen (idExamen, fecha, nota)

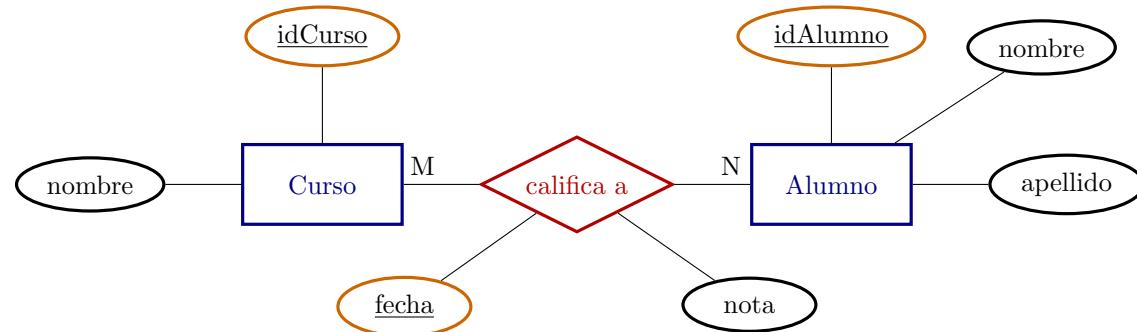
Curso (idCurso, nombre)

Alumno (idAlumno, nombre, apellido)

califica a(idAlumno, idExamen, idCurso)

b)

DER

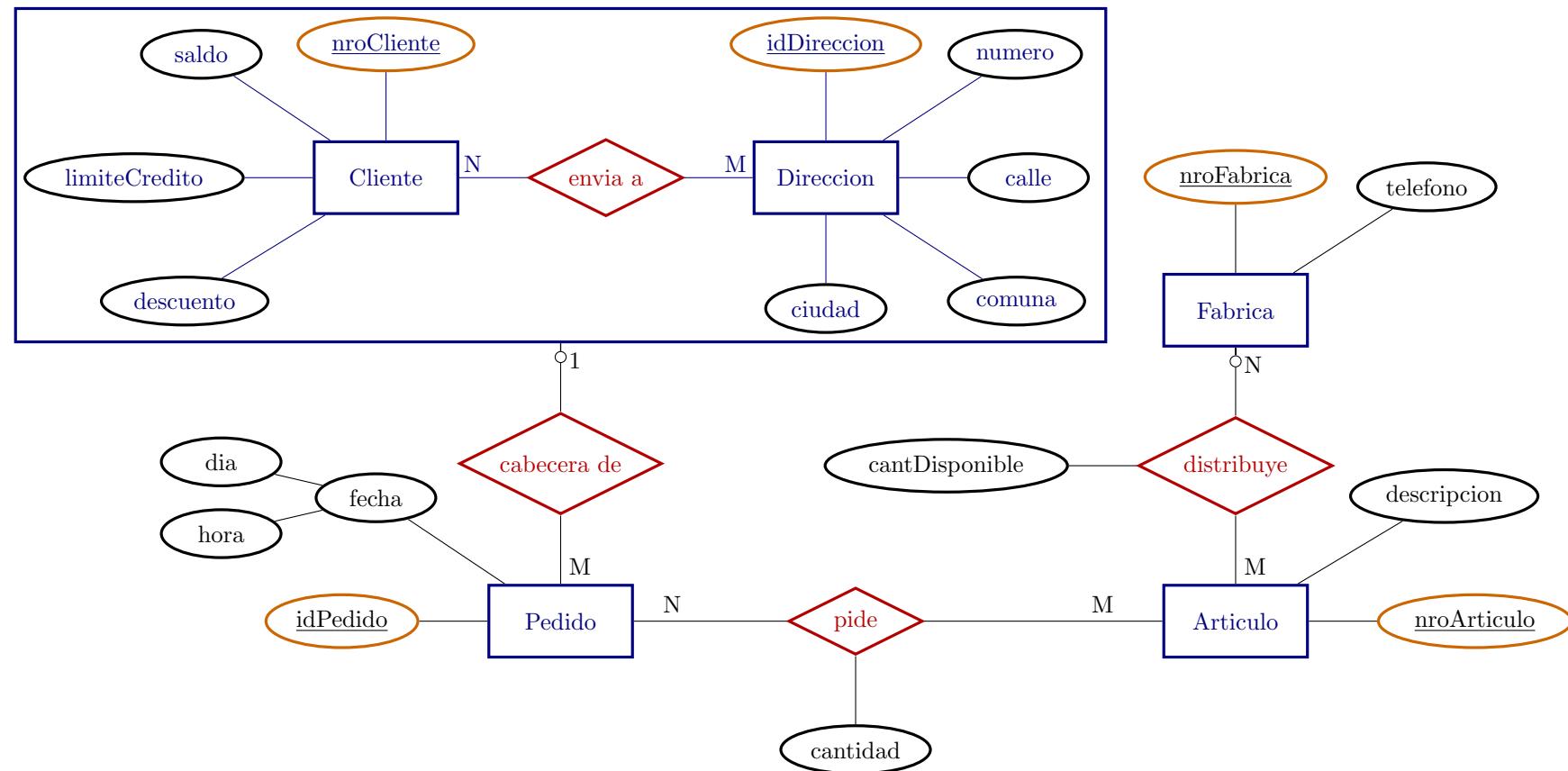


MER

Examen (idExamen, fecha, nota)  
Curso (idCurso, nombre)  
califica a(idAlumno, idCurso, fecha, nota)

## Ejercicio 1.5

DER



Restricciones adicionales:

- *Cliente.limiteCredito* debe ser menor a 3.000.000

**MER**

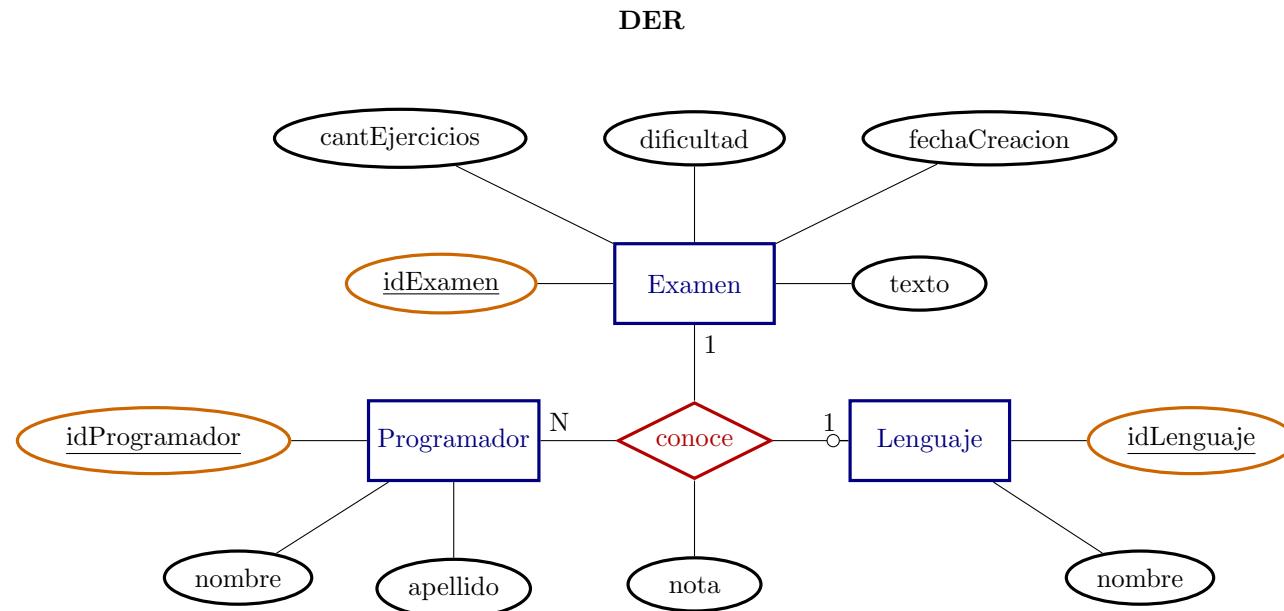
Cliente(nroCliente, saldo, limiteCredito, descuento)  
Direccion(idDireccion, numero, calle, comuna, ciudad)  
enviaA(nroCliente, idDireccion)  
Pedido(idPedido, dia, hora, nroCliente, idDireccion)  
Articulo(nroArticulo, descripcion)  
pide(idPedido, nroArticulo, cantidad)  
Fabrica(nroFabrica, telefono)  
distribuye(nroFabrica, nroArticulo, cantDisponible)

**Notas**

- Al realizar una agregación de una relación M-N, se la toma como a una entidad más. Puede participar en cualquier tipo de relación y cuantas relaciones sea necesario.
- Se pide saber cuantos artículos en total provee la fábrica. No es necesario almacenar este dato, se puede hacer una búsqueda en la tabla *distribuye* de todos los objetos que tienen el *nroFabrica* adecuado y sumar todos los valores.
- Fábrica tiene participación parcial en la relación *distribuye*. Esto quiere decir que habrá algunas que no distribuirán ningún producto a la empresa (estas son las fábricas alternativas).

## Ejercicio 1.6

a)



### MER

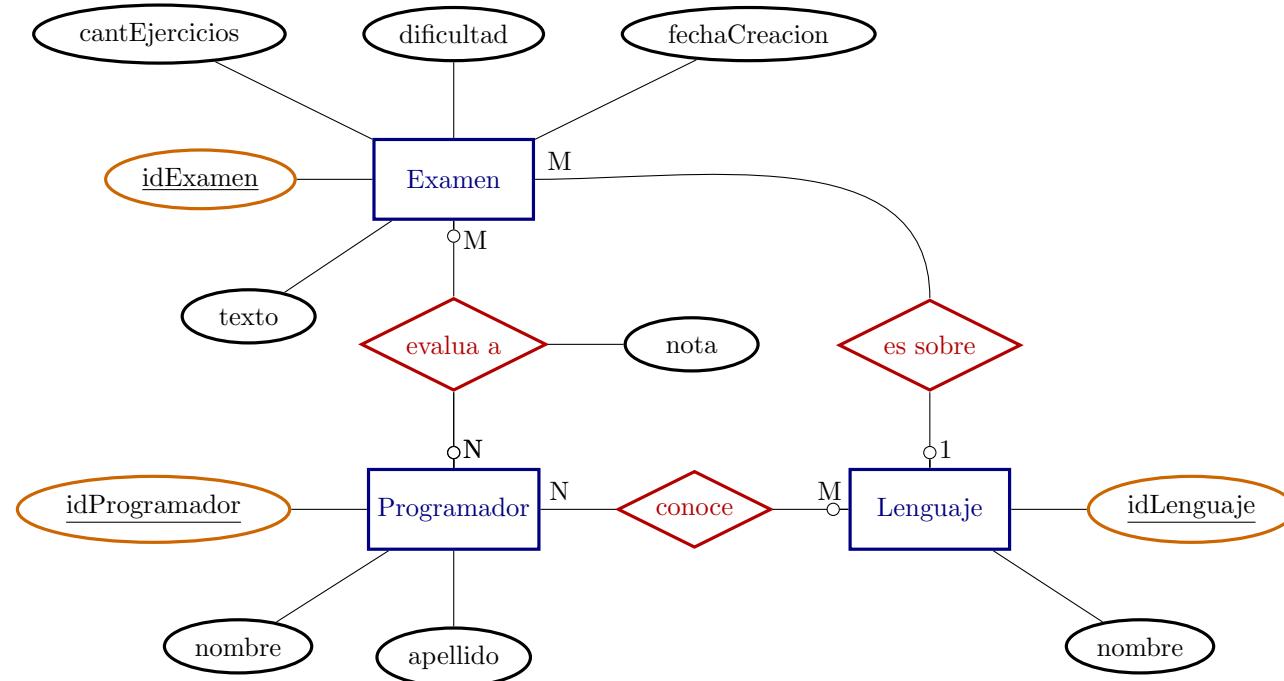
Programador(idProgramador, nombre, apellido)  
 Lenguaje (idLenguaje, nombre)  
 Examen (idExamen, cantEjercicios, dificultad, fechaCreacion, texto)  
 conoce(idProgramador, idLenguaje, examen, nota)

### Notas

- Las relaciones ternarias pueden tener atributos siempre y cuando **no** sean identificatorios

b)

**DER**



**MER**

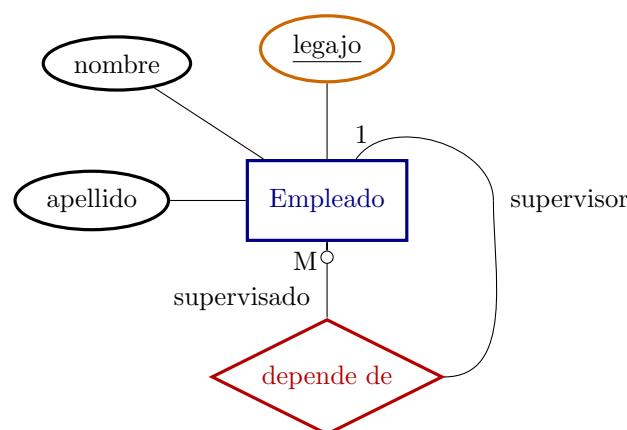
**Programador**(idProgramador, nombre, apellido)  
**Lenguaje** (idLenguaje, nombre)  
**Examen** (idExamen, cantEjercicios, dificultad, fechaCreacion, texto, idLenguaje)  
**conoce**(idProgramador, idLenguaje)  
**evaluaA**(idExamen, idProgramador, nota)

**Restricciones adicionales:**

- Los exámenes que evalúan a un programador deben ser sobre los lenguajes que conoce.

### Ejercicio 1.7 - Discutir inciso e

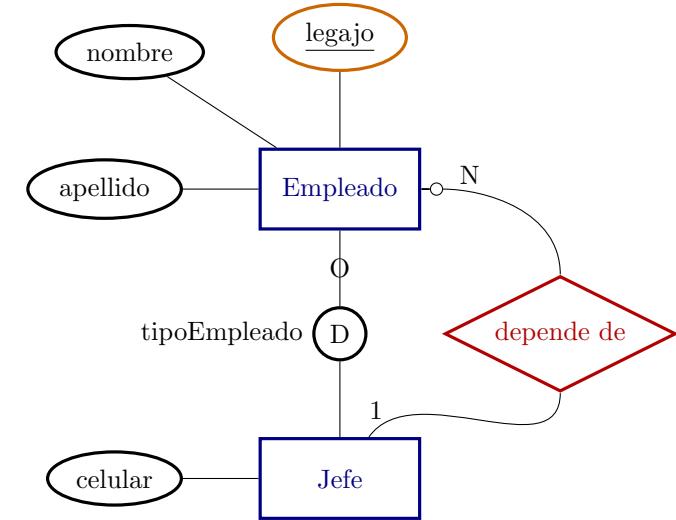
a)



## Restricciones

- Un empleado no puede depender de si mismo.

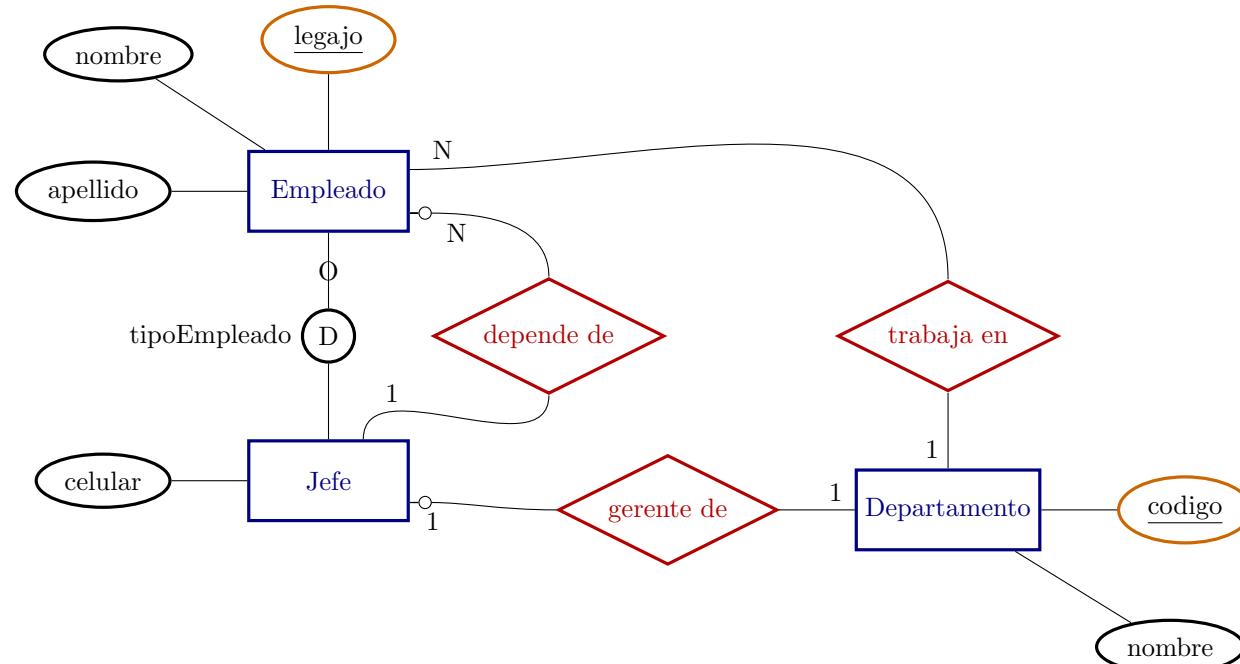
b)



## Restricciones

- Un jefe no puede depender de si mismo.

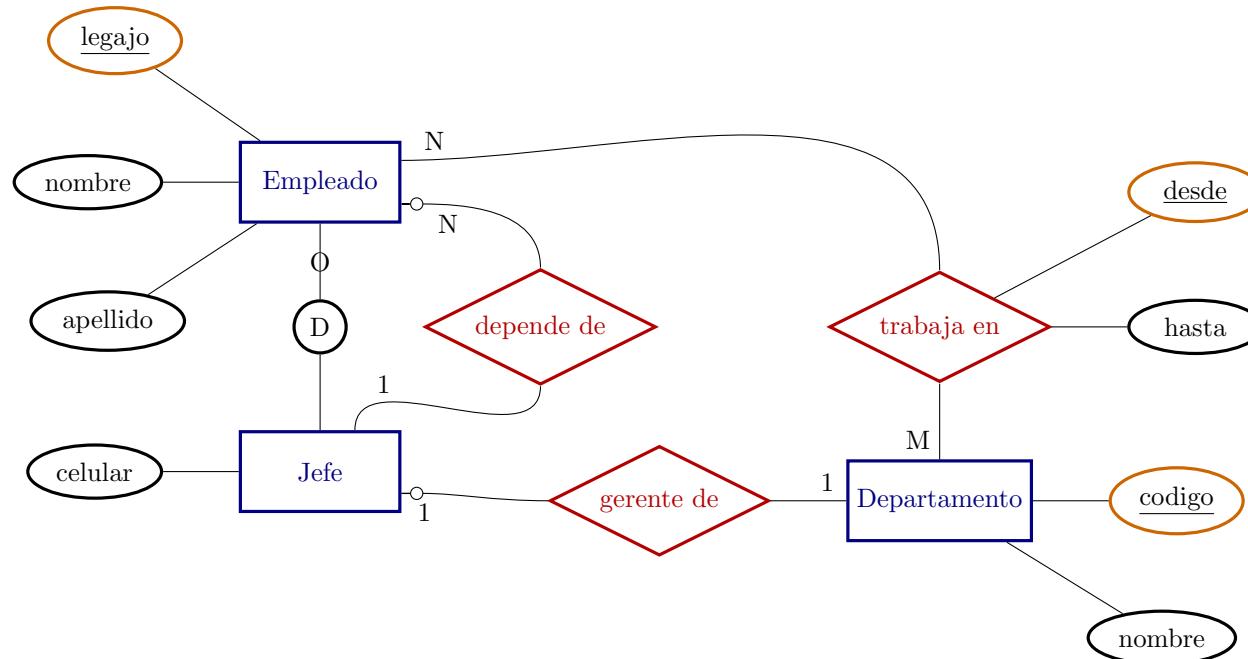
c)



#### Restricciones

- Un jefe no puede depender de si mismo.
- El gerente de un departamento debe trabajar en ese departamento.

d)

**Restricciones**

- Un jefe no puede depender de si mismo.
- El gerente de un departamento debe trabajar en ese departamento.
- Un empleado no puede trabajar en dos departamentos distintos en el mismo intervalo de tiempo (o en intervalos que se superponen).
- Para un empleado que actualmente esté trabajando en un departamento, el atributo *trabajaEn.hasta* será nulo.

e) El modelo propuesto no puede garantizar que se cumpla esta condición. Se podría agregar una Entidad puesto y crear una ternaria con Departamento y Empleado, pero perderíamos el historial de trabajo. En este caso, la relación *es gerente* desaparecería y se podría conseguir esa información consiguiendo todos los empleados que tienen una entrada con el puesto de gerencia en la nueva relación.

**Notas** La relación *trabajaEn* tiene un atributo *hasta* que será null en varios casos. ¿Es necesario arreglar esto? Se podría tener una relación especial sin atributos para los trabajadores actuales y otra para el historial. Pero hay que ver como se guardaría la fecha en la que empezó a trabajar y, cuando alguien termina de trabajar, habría que actualizar dos tablas.

## 2. Ejercicios Avanzados

### Ejercicio 2.1

#### MER 1

Persona(dni, nombre, direccion, telefono, obraSocial, nroAfiliado, nombreSindicato, fechaIngreso, puesto)  
tieneHijos(dni1, dni2)

**Restricción:** Una persona no puede ser su propio hijo ( $dni1 \neq dni2$ )

#### MER2

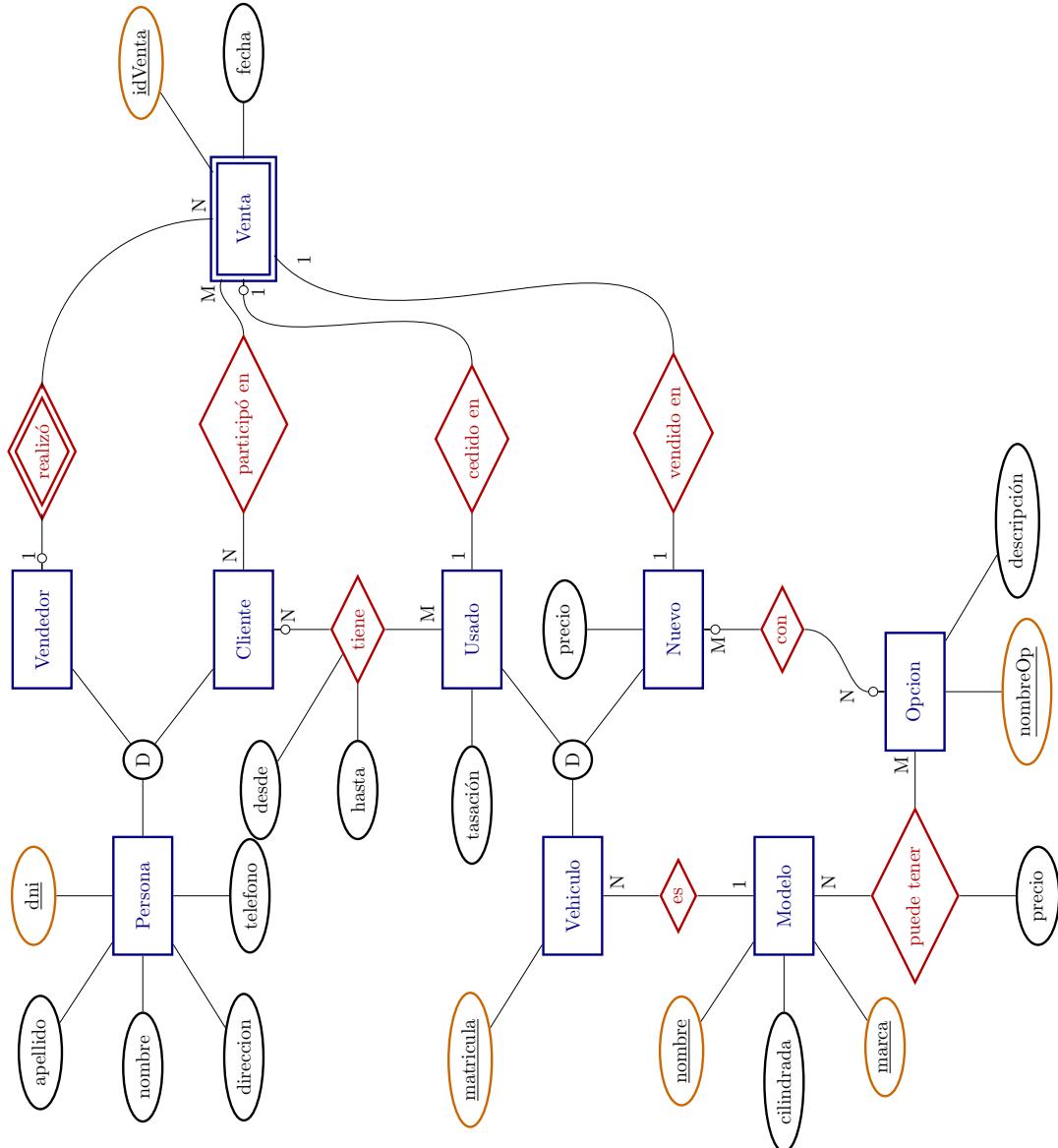
Persona(dniPersona, nombre, direccion, telefono, obraSocial, nroAfiliado, nombreSindicato, fechaIngreso, puesto)  
Hijos(dniHijo, nombre, direccion)  
tiene(dniPersona, dniHijo)

#### MER3

Persona(dni, nombre, direccion, tipo)  
Empleado(dni, nombreSindicato, nroAfiliado, obraSocial, telefono, fechaIngreso, puesto)  
Hijos(dni)  
tiene(dni1, dni2)

**Restriccion:**  $dni1$  debe ser el dni de un empleado y  $dni2$  el dni de un hijo. ( $dni1$  debe tener una entrada en la tabla empleado y  $dni2$  una en la tabla hijo)

## Ejercicio 2.2 - Revisar



### MER

Persona(dni, nombre, apellido, dirección, teléfono, tipo)

Venta(idVenta, fecha, dni, matricula)

Vehiculo(matricula, tipo, nombre, marca)

Usado(matricula, tasacion, idVenta)

Nuevo(matricula, precio)

Modelo(nombre, marca, cilindrada)

Opcion(nombreOp, descripción)

participoEn(dni, idVenta)

tiene(dni, matricula, desde, hasta)

puedeTener(nombre, marca, nombreOp, precio)

con(matricula, nombreOp)

### Restricciones

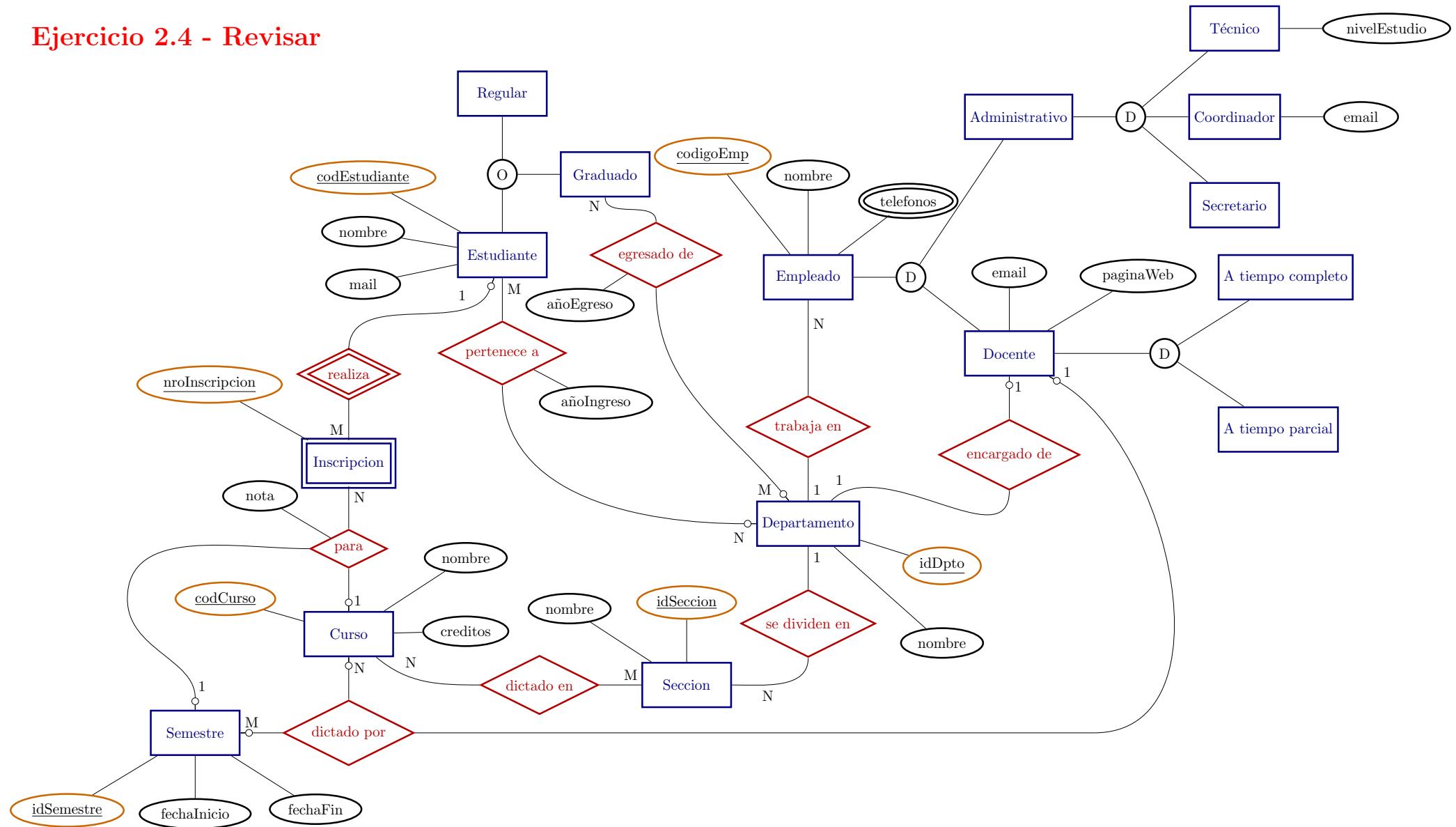
- *pedido.matricula* debe ser un auto usado y debe pertenecer al cliente que participó en la venta *pedido.idVenta*.
  - *tiene.hasta* debe ser menor o igual a la fecha en la que ocurrió la venta en la que el cliente cedió el auto.
  - *vendidoEn.matricula* debe ser la matrícula de un auto nuevo.
  - *nuevo.precio* debe tener en cuenta el precio de cada opción agregada al auto.
- Nota**
- Acá no puedo usar una ternaria Usuario - Usado - Venta porque hay ventas en las que el usuario no cede ningún auto. Si pusiésemos una ternaria, entonces tendríamos una tabla en la que un atributo quedaría nulo la mayoría de las veces.

## Ejercicio 2.3

### MER

Producto(idProducto, descripcion)  
Distribuidor(numDistribuidor, nombre)  
Area(idArea, nombre)  
Local(numLocal, direccion)  
Deposito(numDeposito, Local capacidad)  
Empleado(CUIL, fechaIngreso, tipoContratacion)  
EmpleadoEfectivo( CUIL, cantHijos)  
EmpleadoEfectConGremio(CUIL, fechaAfiliacionGremio)  
EmpleadoEfectConPrepaga(CUIL, numAfiliado)  
trabajaEn(numLocal, CUIL, idArea, cantHoras)  
distribuidoPor(idProducto, numDistribuidor, idArea)

## Ejercicio 2.4 - Revisar



**MER**

Departamento(idDpto, nombre, codigoEmpleado)  
 Empleado(codigoEmpleado, nombre, idDpto)  
 Telefonos(codigoEmpleado, telefono, tipoEmpleado)  
 Docente(codigoEmpleado, email, paginaWeb, tipoDocente)  
 Administrativo(codigoEmpleado, tipoAdministrativo)  
 Tecnico(codigoEmpleado, nivelEstudio)  
 Coordinador(codigoEmpleado, email)  
 Estudiante(codEstudiante, nombre, mail)  
 Regular(codEstudiante)  
 Egresado(codEstudiante)  
 Inscripcion(codEstudiante, nroInscripcion)  
 Curso(codCurso, nombre, creditos)  
 Semestre(idSemestre, fechaInicio, fechaFin)  
 Seccion(idSeccion, nombre, idDpto)  
 egresadoDe(codigoEstudiante, idDpto)  
 para(codEstudiante,nroIncripcion, codCurso, idSemestre, nota)  
 dictadoPor(codCurso, idSemestre, codigoEmp)  
 dictadoEn(curso,idSeccion)

**Restricciones adicionales:**

- Todos los *codigoEstudiante* en *egresadoEn* son estudiantes graduados
- *codEmpleado* en *Departamento* es el código de un docente.
- *codEmpleado* en *dictadoPor* es el código de un docente.
- El docente que dicta un curso debe pertenecer a alguno de los departamentos en donde se dicta.
- Un curso puede ser dictado en una única sección de un departamento. Osea, si hay dos secciones que dictan el mismo curso, entonces son de distinto departamentos.

**Consultas**

- (a) Se pueden conseguir todas las inscripciones realizadas por el alumno *A* usando la relación *para* y agruparlas por semestre.
- (b) Se puede calcular a partir de la relación *para*, consiguiendo todas las notas del alumno.
- (c) Se consiguen todas las inscripciones del alumno usando la relación *para*. Para cada par (*semestre*, *curso*) devuelto, se consigue la entrada correspondiente en *dictadoPor*. De acá, conseguimos el docente que dictó cada curso y, de él, el departamento en el que trabaja.  
Se filtran todos los cursos que corresponden al departamento *D*.
- (d) Idem (c) pero se devuelven todos los departamentos conseguidos (eliminando repetidos).
- (e) Se buscan las inscripciones del alumno *A* que corresponden al curso *C* en la relación *para*. De aquí se consigue los semestres en los que cursó. Usando la relación *dictadoPor* se consigue el docente y, de ahí, el departamento correspondiente.
- (f) Se buscan todas las inscripciones correspondientes al curso *C* en el semestre deseado y calculamos el promedio.  
Si es necesario chequear que el curso pertenece a la sección *S*, entonces solo hay que ver que *S* pertenezca al departamento que la está dictando ese semestre.
- (g) Se consigue filtrando todas las entradas que corresponden al semestre y al profesor indicado en la relación *dictadoPor*.

**Notas**

- En una de los puntos aparece una sección salvaje que no se aclara bien lo que es, por ahora, asumo que son las áreas de cada departamento, por ejemplo, en Computación, tenemos Algoritmos, Sistemas y Teoría.
- Por como hice el modelo, un curso puede ser dictado por un único departamento en cada cuatrimestre.

## Ejercicio 2.5

MER

Consultas

Restricciones adicionales:

