

**Objetivo del ejercicio:**

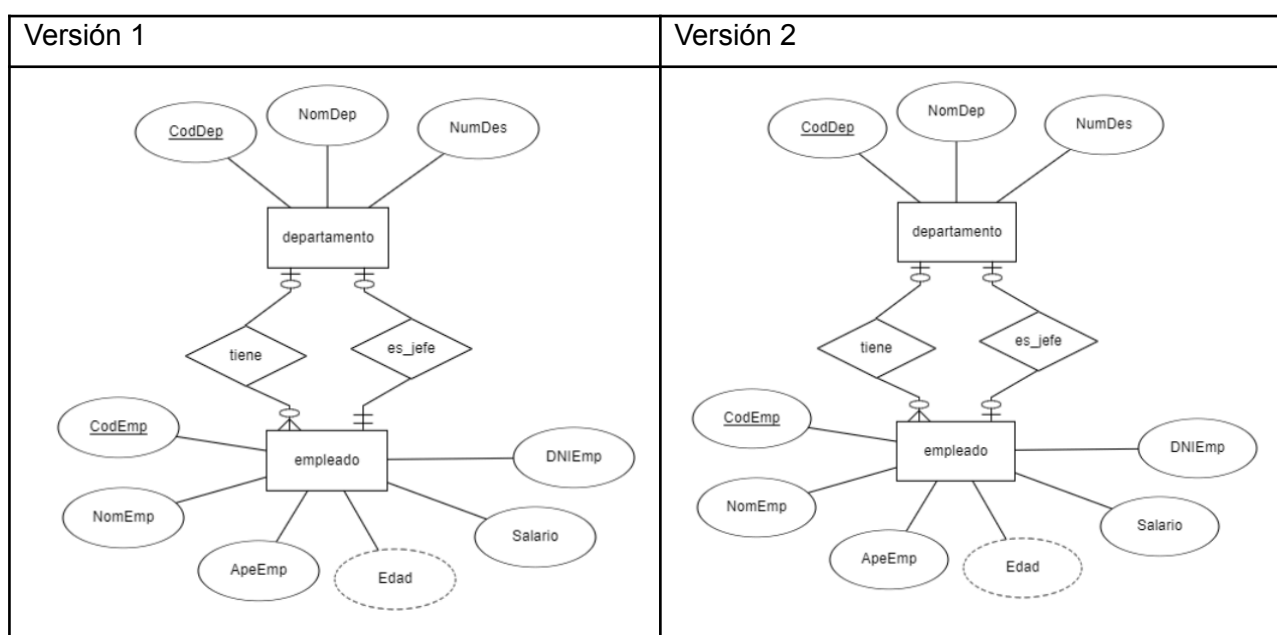
1. Repasar el tipo de correspondencia el tipo de correspondencia 1 a 0 y el atributo derivado.
2. Aprender que entre dos entidades puede existir más de una relación. En este caso, una relación tiene el tipo de correspondencia 1 a n, y la otra relación es de tipo de correspondencia 1 a 0. En la primera versión del ejercicio, en la que todos los departamentos deben tener asignado un jefe, la FK del tipo de correspondencia 1 a 0, NO va a admitir nulos. En la segunda versión, en la que el departamento permite tener durante algún tiempo el jefe de departamento sin asignar, la FK del tipo de correspondencia 1 a 0, SI va a admitir nulos.

**EJERCICIO 9. Empresa 1**

Se desea diseñar una base de datos para gestionar la información sobre los empleados de una empresa, a partir de las siguientes condiciones: para cada empleado dispondremos de su DNI, nombre, edad, salario y departamento en el que trabaja. Un empleado puede no tener asignado un departamento durante un tiempo. De cada departamento sabemos su código, nombre, el número del despacho en el que se ubica y conocemos cuál es el jefe de dicho departamento. Un empleado no podrá ser jefe de más de un departamento. Podemos tener registrado un departamento sin que aún haya empleados en el mismo. Hacer dos versiones del ejercicio. Una primera versión, en la que todos los departamentos deben tener asignado un jefe. Y otra, en la que el departamento permita tener durante algún tiempo el jefe de departamento sin asignar.

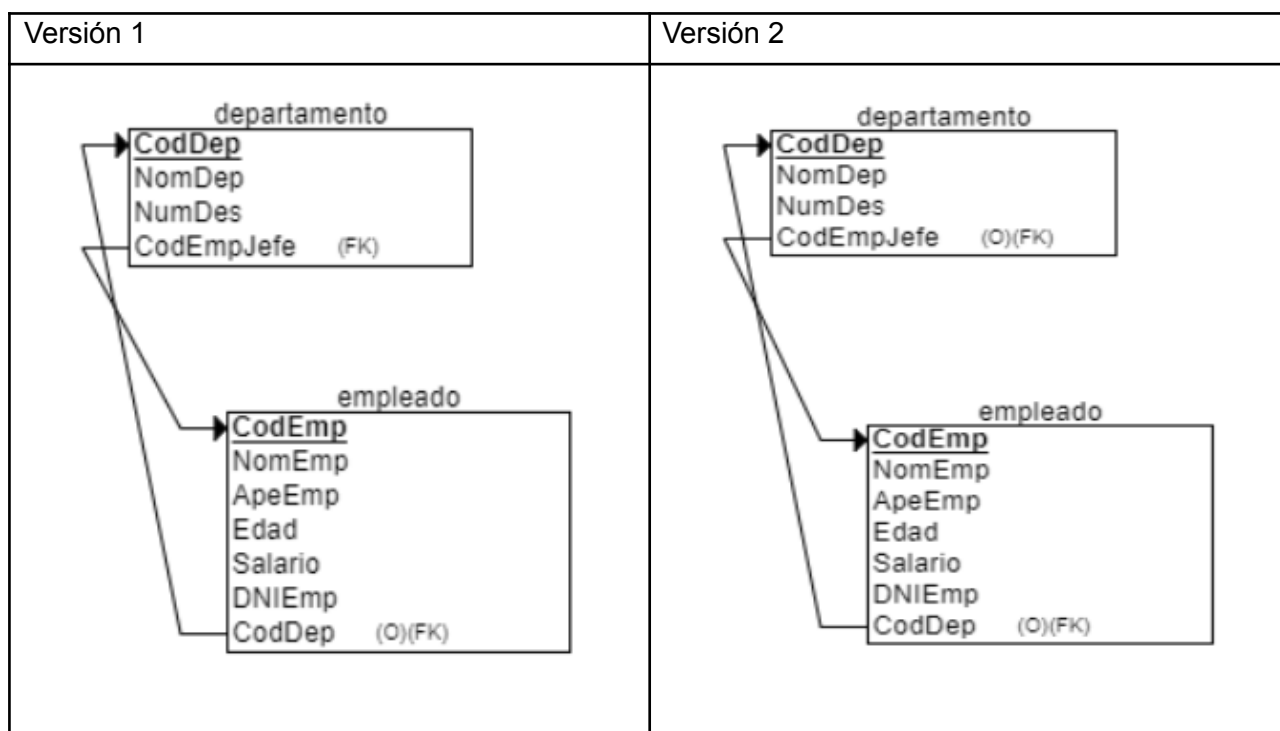
**Se pide:**

1. Modelar la base de datos. Para ello haremos:
  - a. Diseño Conceptual de Datos utilizando un Diagrama o Modelo Entidad-Relación. Lo hacemos en papel y lo pasamos a la Herramienta CASE ERD Plus.

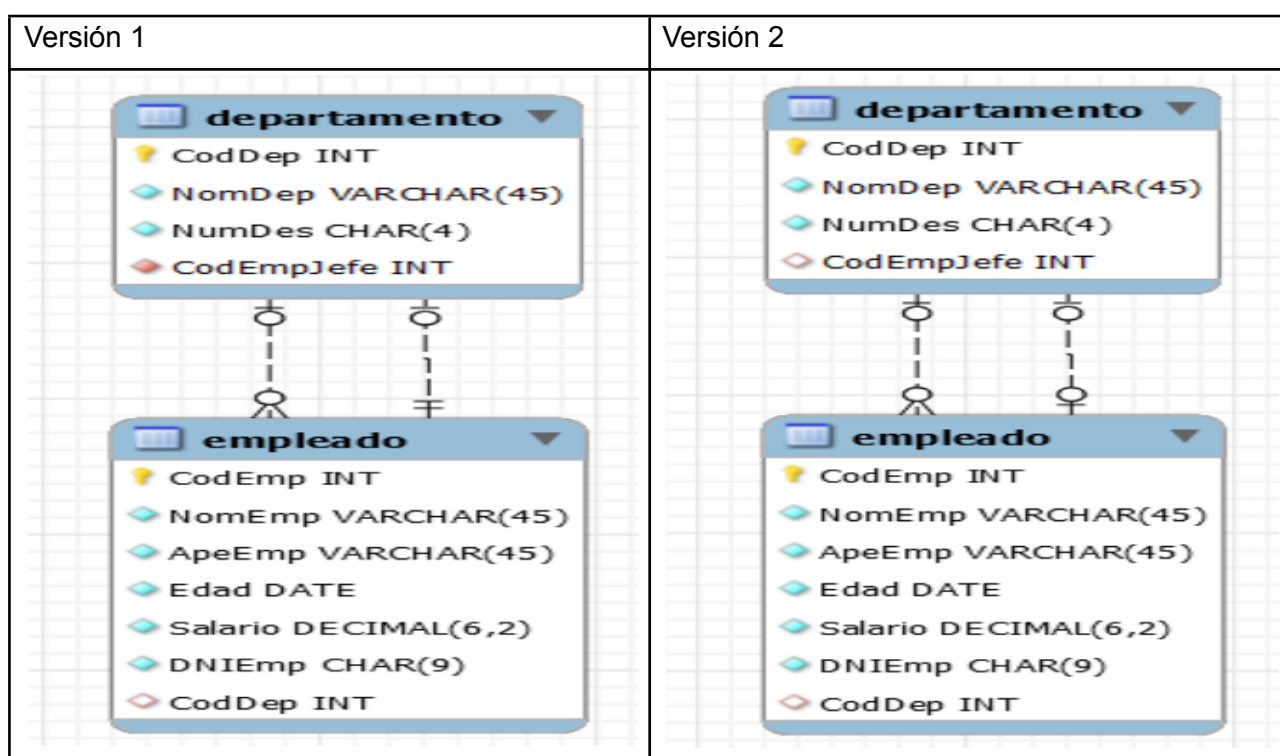


- b. Diseño Lógico de Datos utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench. En este apartado también vamos a poner el Diagrama Referencial que genera ERD Plus a partir del Modelo Entidad-Relación. Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

Diagrama Referencial



DED



- c. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL.  
Si en phpmyadmin intentas ejecutar este código, en cualquiera de las dos versiones observarás que te da error.

Versión 1	Versión 2
<pre>CREATE TABLE departamento (   CodDep INT NOT NULL,   NomDep VARCHAR(45) NOT NULL,   NumDes CHAR(4) NOT NULL,   <b>CodEmpJefe INT NOT NULL,</b>   PRIMARY KEY (CodDep),   FOREIGN KEY (CodEmpJefe) REFERENCES empleado(CodEmp) );  CREATE TABLE empleado (   CodEmp INT NOT NULL,   NomEmp VARCHAR(45) NOT NULL,   ApeEmp VARCHAR(45) NOT NULL,   Edad DATE NOT NULL,   Salario DECIMAL(6,2) NOT NULL,   DNIEmp CHAR(9) NOT NULL,   CodDep INT,   PRIMARY KEY (CodEmp),   FOREIGN KEY (CodDep) REFERENCES departamento(CodDep) );</pre>	<pre>CREATE TABLE departamento (   CodDep INT NOT NULL,   NomDep VARCHAR(45) NOT NULL,   NumDes CHAR(4) NOT NULL,   <b>CodEmpJefe INT NULL,</b>   PRIMARY KEY (CodDep),   FOREIGN KEY (CodEmpJefe) REFERENCES empleado(CodEmp) );  CREATE TABLE empleado (   CodEmp INT NOT NULL,   NomEmp VARCHAR(45) NOT NULL,   ApeEmp VARCHAR(45) NOT NULL,   Edad DATE NOT NULL,   Salario DECIMAL(6,2) NOT NULL,   DNIEmp CHAR(9) NOT NULL,   CodDep INT,   PRIMARY KEY (CodEmp),   FOREIGN KEY (CodDep) REFERENCES departamento(CodDep) );</pre>

El motivo es que cuando intentas crear la tabla de departamento, no lo permite, porque dicha tabla hace referencia a la tabla empleado y aún no existe. Si intentas crear la tabla de empleado en primer lugar, ocurre exactamente lo mismo, no lo permite porque hace referencia a la tabla departamento y esta aún no está creada.

Podemos solucionar de la siguiente manera:

1. Creamos la tabla departamento sin la restricción de FK.
2. Creamos la tabla empleado.
3. Modificamos la estructura de la tabla departamento, añadiendo la restricción de FK.

El código quedaría de la siguiente manera para cada versión:

Versión 1	Versión 2
<pre>CREATE TABLE departamento (   CodDep INT NOT NULL,   NomDep VARCHAR(45) NOT NULL,   NumDes CHAR(4) NOT NULL,   <b>CodEmpJefe INT NOT NULL,</b>   PRIMARY KEY (CodDep) );  CREATE TABLE empleado (   CodEmp INT NOT NULL,   NomEmp VARCHAR(45) NOT NULL,   ApeEmp VARCHAR(45) NOT NULL,   FechNac DATE NOT NULL,   Salario DECIMAL(6,2) NOT NULL,   DNIEmp CHAR(9) NOT NULL,   CodDep INT,   PRIMARY KEY (CodEmp),   FOREIGN KEY (CodDep) REFERENCES departamento(CodDep) );  ALTER TABLE departamento ADD CONSTRAINT fk_jefe FOREIGN KEY (CodEmpJefe) REFERENCES empleado(CodEmp);</pre>	<pre>CREATE TABLE departamento (   CodDep INT NOT NULL,   NomDep VARCHAR(45) NOT NULL,   NumDes CHAR(4) NOT NULL,   <b>CodEmpJefe INT NULL,</b>   PRIMARY KEY (CodDep) );  CREATE TABLE empleado (   CodEmp INT NOT NULL,   NomEmp VARCHAR(45) NOT NULL,   ApeEmp VARCHAR(45) NOT NULL,   FechNac DATE NOT NULL,   Salario DECIMAL(6,2) NOT NULL,   DNIEmp CHAR(9) NOT NULL,   CodDep INT,   PRIMARY KEY (CodEmp),   FOREIGN KEY (CodDep) REFERENCES departamento(CodDep) );  ALTER TABLE departamento ADD CONSTRAINT fk_jefe FOREIGN KEY (CodEmpJefe) REFERENCES empleado(CodEmp);</pre>

Observa que en este caso le hemos puesto el nombre `fk_jefe` a la restricción de FK. Se podría haber obviado `CONSTRAINT fk_jefe`, en ese caso el propio sistema le asigna un nombre.

2. Insertar datos desde phpmyadmin utilizando la sentencia `INSERT INTO` del LMD de SQL.

Versión 1.

En esta versión, primero deberemos introducir algunos empleados, por lo menos los que sean jefe de departamento. Esto es porque un empleado se puede introducir sin especificar a qué departamento pertenece. Sin embargo, un departamento no se puede dar de alta sin especificar cuál es su jefe.

Versión 2.

En esta versión, podemos introducir primero los departamento (esto es porque ahora `CodEmpJef` admite nulo). Después introducimos los empleados. Y a continuación, podremos indicar cuáles son jefes haciendo `UPDATE` en la tabla departamento.