

*Objetivo del ejercicio:*

*1. Aprender el tipo de correspondencia 1 a 1.*

Se quiere diseñar una base de datos para almacenar información sobre los empleados de un gimnasio. De los empleados se quiere almacenar el nombre y los apellidos. Además, de cada empleado se quiere almacenar una foto, la formación académica y la experiencia profesional.

**empleado** → nombre, apellidos

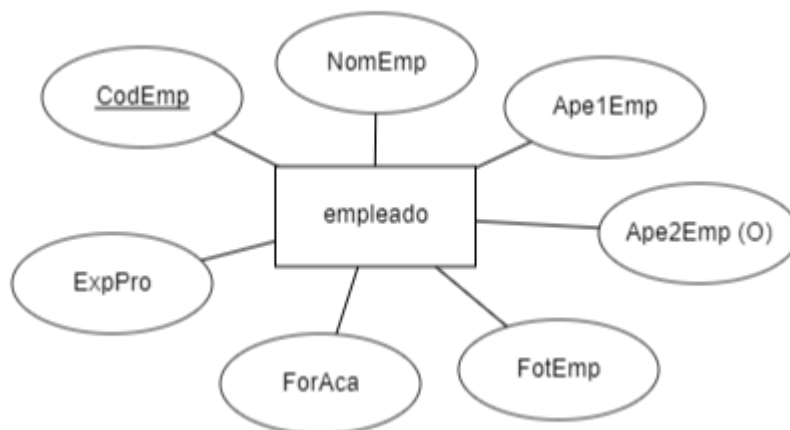
**curriculum** → foto, formación académica y experiencia profesional

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.

Hemos detectado dos entidades, pero sabemos que un empleado tiene solo un curriculum como máximo y como mínimo, y que un curriculum pertenece a un solo empleado como máximo y como mínimo. Es decir, una ocurrencia de una entidad se relaciona con una sola ocurrencia de la otra entidad y viceversa. Tenemos por lo tanto, un tipo de correspondencia 1 a 1. Podemos modelarlo de tres manera diferentes:

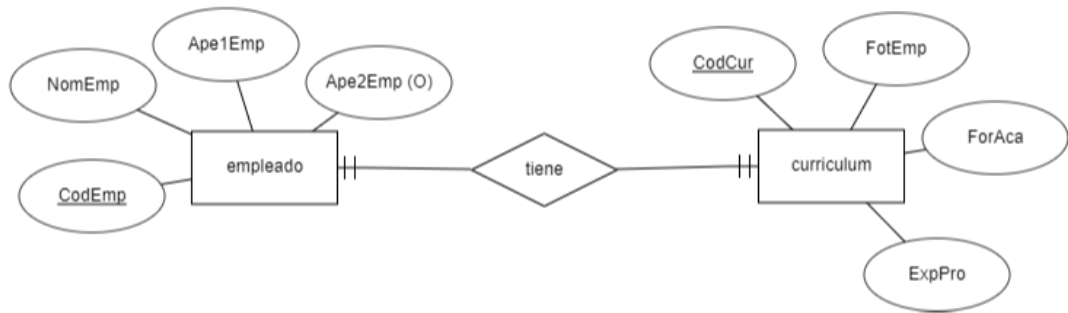
**1 a 1 caso1.** Decidimos dejar todo en una sola tabla.



Se utiliza cuando queremos mostrar siempre la información en conjunto (ej. queremos saber si el empleado es capaz de realizar ciertas funciones para cubrir a otro empleado). Es decir, siempre que accedemos a los datos del empleado, quiero acceder a los datos de su curriculum.

## EJERCICIO 6

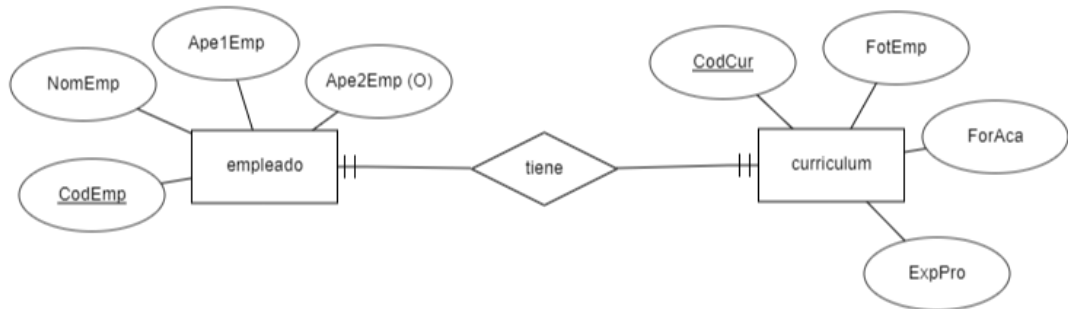
**1 a 1 caso 2.** Decidimos separar la información en dos tablas. CodEmp pasa como FK a curriculum.



Decidimos separar la información en dos tablas porque siempre que accedo a los datos del empleado, no necesito acceder a los datos de su curriculum (igual para el caso 3).

En este diseño y el siguiente, es importante cómo se conectan las entidades, si se conecta de la derecha a la izquierda, la FK quedará en la izquierda, y viceversa.

**1 a 1 caso 3.** Decidimos separar la información en dos tablas. CodCur pasa como FK a empleado.



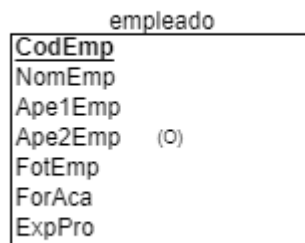
## EJERCICIO 6

- 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.

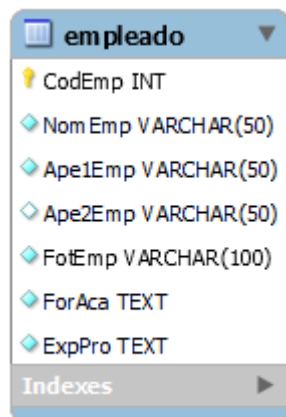
Recuerda que el Diseño Lógico de Datos es hacer el modelo relacional y para ello podemos hacer un DED o un Diagrama Referencial.

### 1 a 1 caso1

#### Diagrama Referencial

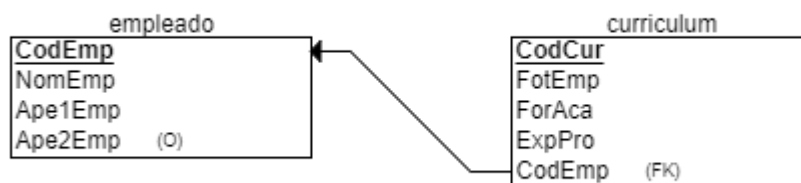


#### DED



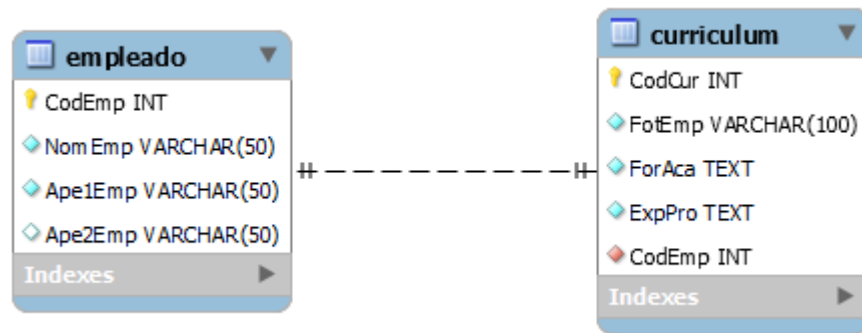
### 1 a 1 caso2

#### Diagrama Referencial



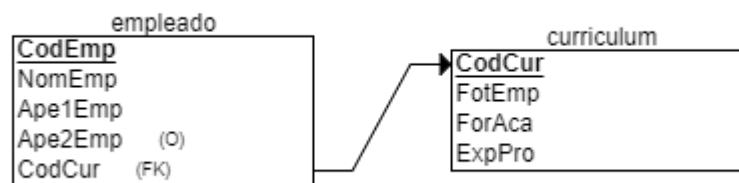
También podemos dejar que CodEmp sea PK además de FK, en lugar de tener CodCur como PK (esta no existiría), ya que igualmente CodEmp no se va a repetir.

DED

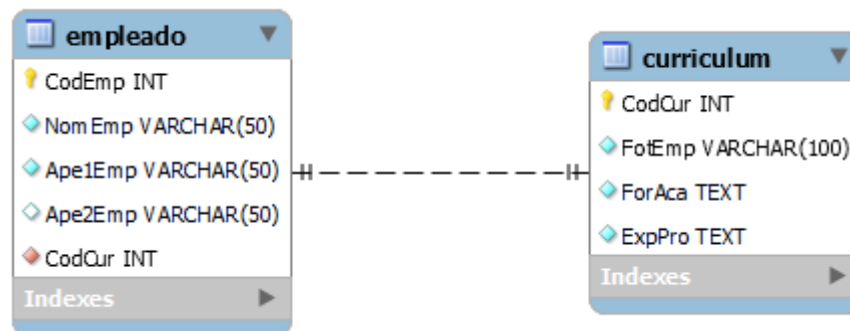


**1 a 1 caso3**

Diagrama Referencial



DED



- c. Diseño Físico de Datos. Creamos la base de datos y las tablas en SQL para los tres casos.

**1 a 1 caso1**

```
CREATE DATABASE Ejercicio6_caso1;
```

```
CREATE TABLE empleado
(
    CodEmp INT NOT NULL,
    NomEmp VARCHAR(50) NOT NULL,
    Ape1Emp VARCHAR(50) NOT NULL,
    Ape2Emp VARCHAR(50),
    FotEmp VARCHAR(100) NOT NULL,
```

## EJERCICIO 6

```
ForAca TEXT NOT NULL,  
ExpPro TEXT NOT NULL,  
PRIMARY KEY (CodEmp)  
);
```

A las fotografías se le indica VARCHAR porque se va a añadir la ruta donde se almacene la foto.

### 1 a 1 caso2

```
CREATE DATABASE Ejercicio6_caso2;
```

```
CREATE TABLE empleado  
(  
    CodEmp INT NOT NULL,  
    NomEmp VARCHAR(50) NOT NULL,  
    Ape1Emp VARCHAR(50) NOT NULL,  
    Ape2Emp VARCHAR(50),  
    PRIMARY KEY (CodEmp)  
);
```

```
CREATE TABLE curriculum  
(  
    FotEmp VARCHAR(100) NOT NULL,  
    ForAca TEXT NOT NULL,  
    ExpPro TEXT NOT NULL,  
    CodCur INT NOT NULL,  
    CodEmp INT NOT NULL,  
    PRIMARY KEY (CodCur),  
    FOREIGN KEY (CodEmp) REFERENCES empleado(CodEmp)  
);
```

### 1 a 1 caso3

```
CREATE DATABASE Ejercicio6_caso3;
```

```
CREATE TABLE curriculum  
(  
    FotEmp VARCHAR(100) NOT NULL,  
    ForAca TEXT NOT NULL,  
    ExpPro TEXT NOT NULL,  
    CodCur INT NOT NULL,  
    PRIMARY KEY (CodCur)  
);
```

```
CREATE TABLE empleado  
(  
    CodEmp INT NOT NULL,  
    NomEmp VARCHAR(50) NOT NULL,  
    Ape1Emp VARCHAR(50) NOT NULL,  
    Ape2Emp VARCHAR(50),  
    CodCur INT NOT NULL,  
    PRIMARY KEY (CodEmp),  
    FOREIGN KEY (CodCur) REFERENCES curriculum(CodCur)  
);
```

);

2. Insertar datos desde phpmyadmin. Los mismos en las tres bases de datos.

### **1 a 1 caso1**

```
INSERT INTO empleado (CodEmp, NomEmp, Ape1Emp, Ape2Emp, FotEmp, ForAca,
ExpPro)
VALUES (1, 'Santino', 'Marrero', 'Quintero', 'C:\\Users\\Usuario\\Pictures\\Curriculum1', 'CFS
Desarrollo de Aplicaciones Web', '2 meses de prácticas'),
      (2, 'Nacho', 'Velasco', 'Diez', 'C:\\Users\\Usuario\\Pictures\\Curriculum2', 'CFS
Desarrollo de Aplicaciones Multiplataforma', '3 años de autónomo'),
      (3, 'Felix', 'Mendizábal', 'Seguí', 'C:\\Users\\Usuario\\Pictures\\Curriculum3', 'Grado
Ingeniería Informática', '5 años como informático'),
      (4, 'Milagros', 'Solera', NULL, 'C:\\Users\\Usuario\\Pictures\\Curriculum4', 'CFS
Desarrollo de Aplicaciones Web', '3 años');
```

Se debe utilizar un carácter de escape '\\' para que inserte la barra a continuación (ya que es el propio carácter de escape y no lo inserta). Este carácter de escape permite que interprete la barra '\\' a continuación (u otro carácter).

### **1 a 1 caso2**

```
INSERT INTO empleado (CodEmp, NomEmp, Ape1Emp, Ape2Emp)
VALUES (1, 'Santino', 'Marrero', 'Quintero'),
      (2, 'Nacho', 'Velasco', 'Diez'),
      (3, 'Felix', 'Mendizábal', 'Seguí'),
      (4, 'Milagros', 'Solera', NULL);
```

```
INSERT INTO curriculum (CodCur, FotEmp, ForAca, ExpPro, CodEmp)
VALUES (1, 'C:\\Users\\Usuario\\Pictures\\Curriculum1', 'CFS Desarrollo de Aplicaciones
Web', '2 meses de prácticas', 1),
      (2, 'C:\\Users\\Usuario\\Pictures\\Curriculum2', 'CFS Desarrollo de Aplicaciones
Multiplataforma', '3 años de autónomo', 2),
      (3, 'C:\\Users\\Usuario\\Pictures\\Curriculum3', 'Grado Ingeniería Informática', '5 años como
informático', 3),
      (4, 'C:\\Users\\Usuario\\Pictures\\Curriculum4', 'CFS Desarrollo de Aplicaciones Web', '3
años', 4);
```

### **1 a 1 caso3**

```
INSERT INTO curriculum (CodCur, FotEmp, ForAca, ExpPro)
VALUES (1, 'C:\\Users\\Usuario\\Pictures\\Curriculum1', 'CFS Desarrollo de Aplicaciones
Web', '2 meses de prácticas'),
      (2, 'C:\\Users\\Usuario\\Pictures\\Curriculum2', 'CFS Desarrollo de Aplicaciones
Multiplataforma', '3 años de autónomo'),
      (3, 'C:\\Users\\Usuario\\Pictures\\Curriculum3', 'Grado Ingeniería Informática', '5 años como
informático'),
      (4, 'C:\\Users\\Usuario\\Pictures\\Curriculum4', 'CFS Desarrollo de Aplicaciones Web', '3
años');
```

```
INSERT INTO empleado (CodEmp, NomEmp, Ape1Emp, Ape2Emp, CodCur)
VALUES (1, 'Santino', 'Marrero', 'Quintero', 1),
        (2, 'Nacho', 'Velasco', 'Diez', 2),
        (3, 'Felix', 'Mendizábal', 'Seguí', 3),
        (4, 'Milagros', 'Solera', NULL, 4);
```

**3. Realizar las siguientes consultas en SQL:**

- Muestra los datos de los empleados en las tres bases de datos.

**1 a 1 caso1**

```
SELECT *
FROM empleado;
```

**1 a 1 caso2**

```
SELECT *
FROM empleado em INNER JOIN curriculum cu ON (em.CodEmp = cu.CodEmp);
```

**1 a 1 caso3**

```
SELECT *
FROM empleado em INNER JOIN curriculum cu ON (em.CodCur = cu.CodCur);
```