

EJERCICIO 1

El objetivo de este primer ejercicio es **mostrar** paso a paso cómo se hace una **SENCILLA** base de datos: **Aprendemos que hay que hacer DCD, DLD y DFD. Y Que es parte del ciclo de desarrollo del software. En concreto el DCD y DLD está en el Análisis y el DFD en el Diseño.**

Supongamos que una empresa desea almacenar y gestionar información de forma automática sobre sus departamentos y empleados.

Se pide:

1. Enumerar los pasos que pensáis que se deben seguir.

- Identificación de necesidades.

Entrevista personal con el responsable de la empresa para darnos la especificación del programa que desean que les desarrollemos.

Elaborar una lista de requisitos. Por ejemplo:

1. Se necesita guardar la siguiente información sobre los empleados:

Nombre y Apellidos, Puesto, DNI, Edad, Antigüedad, Salario Actual, Domicilio, Tfno, email, Tipo de contrato, Horas extras mes. Para simplificar este primer ejercicio nos quedaremos solo con Nombre y Apellidos y DNI.

2. Se necesita guardar la siguiente información sobre los departamentos:

Nombre, Presupuesto anual, Ubicación. Para simplificar este primer ejercicio nos quedaremos solo con Nombre.

3. El programa debe interactuar con el usuario dando respuesta a las siguientes necesidades:

Conocer en general cualquier dato sobre los empleados almacenados.

Conocer para un empleado a que departamento pertenece este.

Conocer en general cualquier dato sobre los departamentos almacenados.

Listados:

Listado con datos de empleados en el que aparezca el nombre del departamento al que pertenece.

Listado con datos de departamentos.

Listado de cada departamento junto con los datos de los empleados que pertenecen a él.

- Análisis.
- Diseño.
- Construcción del Programa.
- Pruebas.
- Instalación.
- Explotación
- Mantenimiento.

EJERCICIO 1

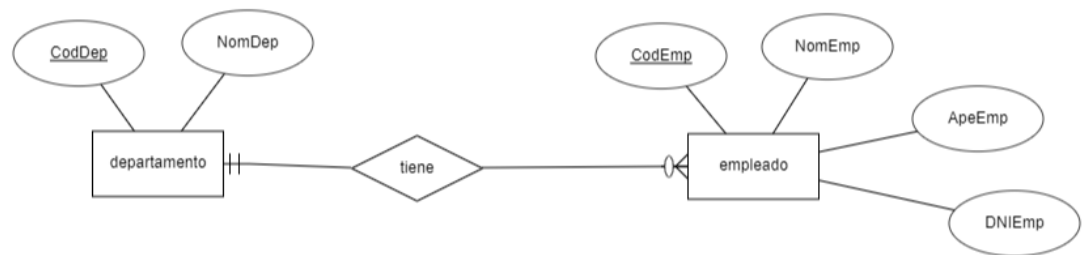
2. Modelar la base de datos. Para ello haremos:

- 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.

<https://erdplus.com/>

Aprendemos el concepto de entidad, atributo, relación, claves candidatas, clave primaria y clave alternativa. Tipo de correspondencia 1 a muchos. Y Cardinalidad máxima y mínima.

Modelo Entidad - Relación



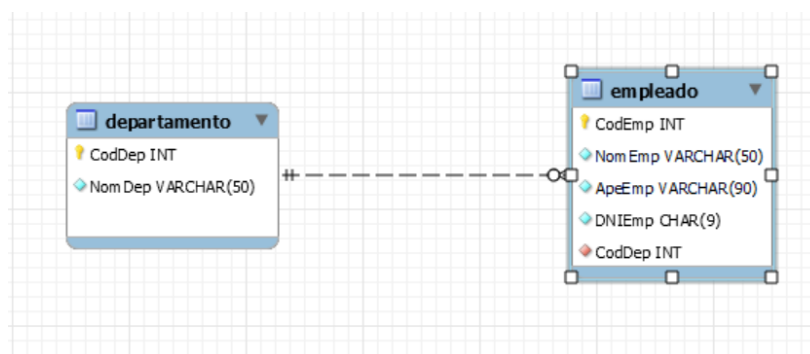
- b. **Diseño Lógico de Datos** utilizando un Diagrama de Estructura de datos (DED). Lo hacemos en papel y lo pasamos a la Herramienta CASE MySql Workbench.

<https://dev.mysql.com/downloads/workbench/>

Aprendemos lo que es una clave foránea FK (clave ajena o clave extranjera)

En un tipo de correspondencia 1 a muchos la FK se pone en la entidad donde va el mucho.

DED



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



c. Diseño Físico de Datos.

Instalamos xampp. <https://www.apachefriends.org/download.html>.

Creamos la base de datos y las tablas en SQL.

```
CREATE DATABASE ejercicio1;
```

```
CREATE TABLE departamento
(
  CodDep INT NOT NULL,
  NomDep VARCHAR(50) NOT NULL,
  PRIMARY KEY (CodDep)
);
```

```
CREATE TABLE empleado
(
  CodEmp INT NOT NULL,
  NomEmp VARCHAR(50) NOT NULL,
  ApeEmp VARCHAR(90) NOT NULL,
  DNIEmp CHAR(9) NOT NULL,
  CodDep INT NOT NULL,
  PRIMARY KEY (CodEmp),
  FOREIGN KEY (CodDep) REFERENCES departamento(CodDep)
);
```

1. Insertar datos desde phpmyadmin.

```
INSERT INTO departamento (CodDep,NomDep)
VALUES (1,'Departamento1'),
       (2,'Departamento2'),
       (3,'Contabilidad'),
       (4,'Nominas'),
       (5,'Atención al cliente');
```

```
INSERT INTO empleado
(CodEmp,NomEmp,ApeEmp,DNIEmp,CodDep)
VALUES (1,'Emp1','Ape11 Ape12','12345678A',1),
       (2,'Emp2','Ape21 Ape22','12345678B',1),
       (3,'Emp3','Ape31 Ape32','12345678C',2),
       (4,'Emp3','Ape41 Ape42','12345678D',2),
       (5,'Emp3','Ape51 Ape52','12345678E',3),
       (6,'Emp3','Ape61 Ape62','12345678F',3);
```

2. Realizar las siguientes consultas en SQL:

- Muestra todas las filas y todos los campos de las tablas

```
SELECT *
```

```
FROM departamento
```

```
SELECT CodDep,NomDep
```

```
FROM departamento
```

```
SELECT *
```

```
FROM empleado;
```

```
SELECT CodEmp, NomEmp, ApeEmp, CodDep,DNIEmp
```

```
FROM empleado;
```

- Muestra algunos campos de las tablas: Obtener el nombre, apellidos y DNI de todos los empleados.

```
SELECT NomEmp,ApeEmp,DNIEmp
```

```
FROM empleado;
```

- Muestra la fecha actual

```
SELECT CURDATE();
```

```
SELECT CURRENT_DATE;
```

- Muestra todos los campos de la tabla EMPLEADOS para aquellas filas que cumplan la condición de que el código de empleado sea mayor que 1:

```
SELECT *
```

```
FROM empleado
```

```
WHERE CodEmp > 1
```

- Muestra todos los campos de la tabla EMPLEADOS para aquellas filas que cumplan la condición de que el código de empleado sea mayor que 1 y que el empleado se llame Emp3.

```
SELECT *
```

```
FROM empleado
```

EJERCICIO 1

WHERE CodEmp > 1 AND NomEmp ='Emp3'

Siempre que comparamos con algo diferente a un número hay que poner comillas simples.

- Muestra el nombre de los empleados, junto con el nombre del departamento en el que se encuentra:

```
SELECT NomEmp, NomDep
```

```
FROM departamento INNER JOIN empleado ON  
(departamento.CodDep=empleado.CodDep)
```

Se puede omitir el INNER.

También podemos usar un alias de tablas, que en este caso ha sido d para departamento y e para empleado.

- Muestra el nombre los departamentos, junto con el nombre de los empleados que pertenecen a cada uno de ellos:

```
SELECT NomDep, NomEmp
```

```
FROM departamento INNER JOIN empleado ON  
(departamento.CodDep=empleado.CodDep)
```

- Obtener nombre y apellidos de los empleados del departamento de RRHH.

```
SELECT NomEmp, ApeEmp
```

```
FROM departamento d INNER JOIN empleado e ON (d.CodDep=e.CodDep)  
WHERE NomDep='RRHH'
```