 CENTRE ESPÈCIFIC D'EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA	CICLO FORMATIVO: CFGS DESARROLLO DE APLICACIONES WEB.	MÓDULO: DESARROLLO WEB DE ENTORNO SERVIDOR. EXAMEN EVALUACIÓN 1	CURSO 19/20 FECHA: Enero 2020	CALIFICACIÓN:
	PROFESOR: Francisco Aldarias Raya.	NOMBRE:		

[MODELO 2]

**Instrucciones:**

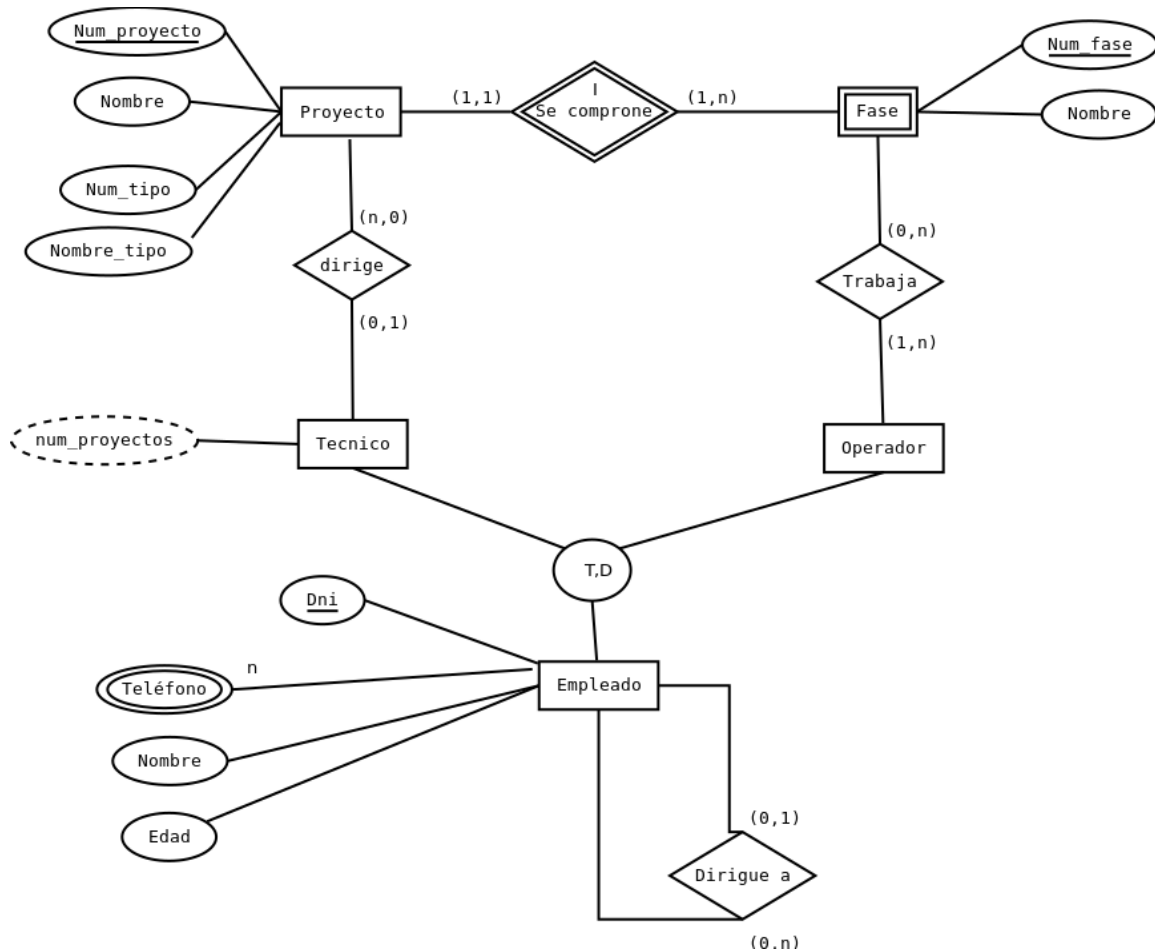
El examen dura 1:30min. Se permite salir después de pasar 20 minutos. Se dará papel del centro y se graparán al final del enunciado del examen al entregarse. Poner el nombre en todas las hojas y numerarlas. Al entregar el examen meterlo en el sobre y poner el nombre en la parte de fuera. No se permite llevar el examen, ni hojas vacías. La nota se enviará por email una vez corregidos todos los exámenes.

**Pregunta 1** (4puntos). Realizar el Diseño conceptual mediante el diagrama Entidad Relación de lo siguiente:

El IES CEED ha decidido para mejorar el servicio de su videoteca hacer una base de datos para almacenar información de las películas que ofrece al sus alumnos. La información que desea almacenar es la siguiente:

1. De toda película se tiene la información siguiente: código, título, nacionalidad, productora y datos de producción ( por ejemplo, Quo Vadis, Estados Unidos, MDB, 1985 )
2. De una película pueden participar diversos actores (nombre, nacionalidad y sexo ), algunos de ellos como actores secundarios y otros como actores principales (protagonistas).
3. De los actores que protagonizan una película, interesa conocer el nombre del personaje que interpretan en la misma.
4. Una película esta dirigida por un director (nombre, nacionalidad y año de nacimiento).
5. De cada película se dispone de uno o más ejemplares diferenciados por un numero de ejemplar y características de sus estado de conservación.
6. Un ejemplar se puede encontrar prestado a algún alumno (nombre, curso, grupo). Se desea almacenar la fecha de comienzo del préstamo y de la devolución
7. Cada alumno puede llevarse prestado como máximo 4 ejemplares.
8. Un alumno ha de ser avalado por otro alumno que responda de le en caso de tener problemas en el préstamo. Cada alumno como máximo puede avalar a otro alumno.
9. Podran haber persona almacenadas que no sean alumno, director o actor. Pero Un tipo determinado de personas puede ser de otro tipo. Por ejemplo un actor puede ser director de la misma película o diferentes.

**Pregunta 2** (3 puntos). Realizar el diseño relacional y el paso a 3ra forma normal del siguiente diagrama ER.



Numerar las relaciones y aplicar hasta la 3fn antes pasar a otro relación. Primero se pondrá la relación sin normalizar y a continuación se normaliza.

Consideraciones:

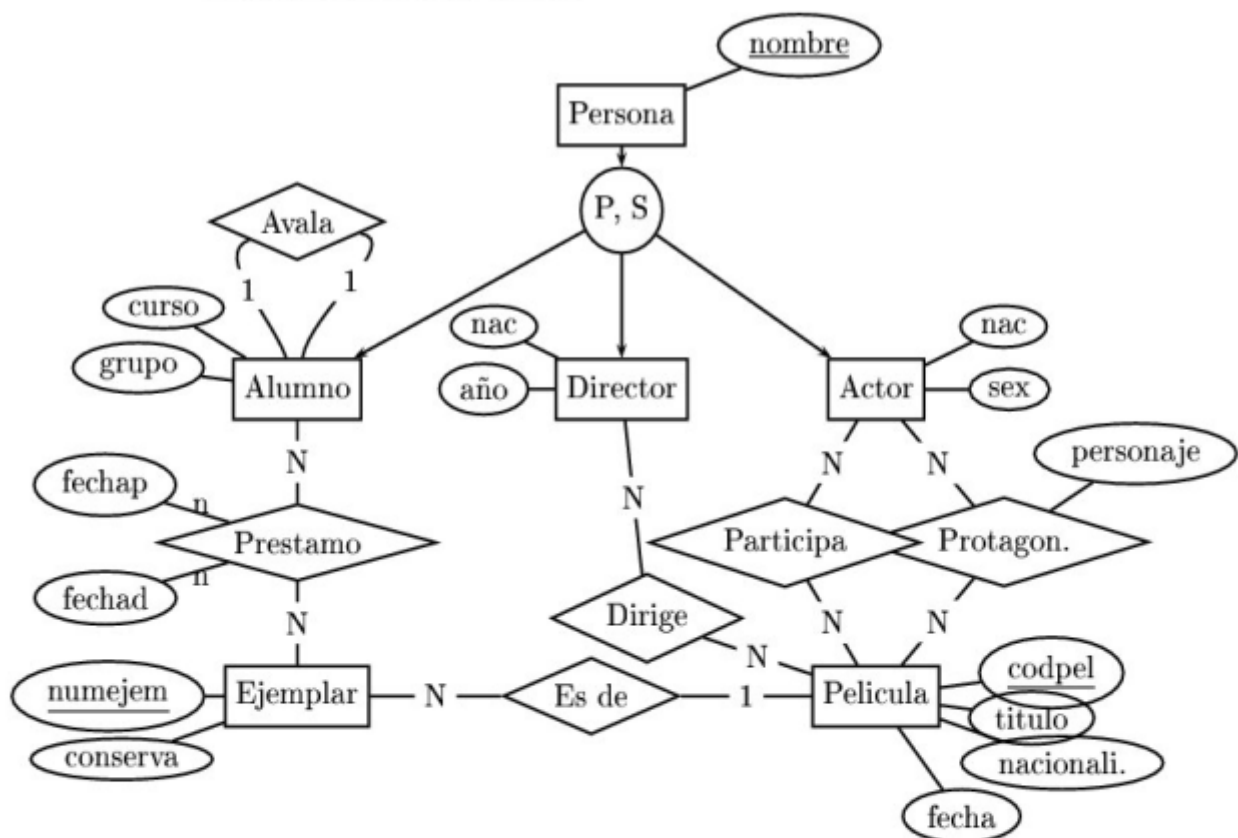
- Un empleado puede tener muchos teléfonos y un teléfono puede ser de muchos empleados.
- Un proyecto pertenece a un tipo, y un tipo puede estar en muchos proyectos.
- El numero de proyectos se calcula a través de la relación entre técnico y proyecto. Es decir, que podemos saber los proyectos dirigidos por un empleados contándolos en la relación con proyecto.
- La edad es un dato que se pide en el momento de la toma de datos. Y no se pide la fecha de nacimiento.
- El numero de fase se repite de un proyecto a otro.

**Pregunta 3** (3puntos). Realizar el diseño físico de la pregunta 2, indicando el SGBD elegido entre los vistos en clase, creando un script para crear las tablas, otro para añadir al menos 2 filas y otro para borrar las tablas. Poner las restricciones de clave primaria y clave ajena como constraint y dándoles un nombre.

Restricciones de clave ajena:

- No se deja borrar técnicos con proyectos.
- Se podrá borrar un operador de una fase, eliminando por tanto la relación existente.
- Un Empleado que dirige a otros no se podrá borrar pero si actualizar en cascada sobre los empleados que dirige.
- Si se borra un empleado se borra sus teléfonos.

Solución  
Pregunta 1.



RI:

- En una película, un actor protagonista, no puede ser no protagonista.
- Un alumno como máximo puede tener 4 prestamos.
- Un alumno no puede avalarse a sí mismo (antireflexiva).
- Un alumno no puede avalar al mismo que le avala (antisimétrica).

Pregunta 2

1. Empleado ( dni, {telefono}, nombre, edad, dirige)

cp: dni

ca: dirige → Empleado

Atributo n-evaluado telefono. No está en 1fn.

[1] 1.1. **Empleado** ( dni, nombre, edad, dirige)

cp: dni

ca: dirige → Empleado

[2] 1.2 **Telefono**( Numero )

cp: numero

[3] 1.3 **TelefonoEmpleado**( Numero, dni )

cp: (numero, dni)

ca: numero → Telefono

ca: dni → Empleado

2. Técnico ( dni , num\_proyectos)

cp: dni

ca: dni → Empleado

[4] 2.1. **Técnico** ( dni)

cp: dni

ca: dni → Empleado

Se elimina num\_proyectos por ser derivado

[5] 3. **Operado** ( dni )

cp: dni

ca: dni → Empleado

[6]4. **Fase** ( num\_fase, nombre, num\_proyecto)

cp: (num\_fase, num\_proyecto)

ca: num\_proyecto → Proyecto

5. Proyecto (num\_proyecto, nombre, num\_tipo, nombre\_tipo, dni\_tecnico)

cp: num:proyecto

ca: dni\_tecnico → Proyecto

vnn: dni\_tecnico

Dependencia Funcional entre atributos no clave: num\_tipo > nombre\_tipo ( no está en 2fn)

[7]5.1 **Proyecto** (num\_proyecto, nombre, num\_tipo, dni\_tecnico)

cp: num:proyecto

ca: dni\_tecnico → Proyecto

ca: num\_tipo → Tipo

vnn: dni\_tecnico

[8] 5.2 **TipoProyecto** (num\_tipo, nombre\_tipo)

cp: num\_tipo

[9] 6. **Trabaja** ( num\_fase, dni\_operador)

cp: num\_fase, dni\_operador

ca: num\_fase → fase

ca: dni\_operador → operador

Salen [9] Relacione en 3FN.

### **Pregunta 3. Para Mysql**

#### **3.1 Creación de tablas.**

```
create table empleado (  
dni varchar(50),  
nombre varchar(50),  
edad numeric,  
dirige varchar2(50) ,
```

```
constraint empleado_pk primary key (dni),
constraint empleado_fk foreign key dirige references empleado(dni) on update cascade
);
```

```
create table telefono (
numero varchar(20),
constraint telefono_pk primary key (numero)
);
```

```
create table telefonoempleado (
numero varchar(20),
dni varchar(50),
constraint telefonoempleado_pk primary key ( numero, dni)
constraint telemp_fk_tel foreign key telefono(numero) on delete cascade;
constraint telemp_fk_emp foreign key empleado(dni) on delete cascade;
);
```

```
create table Tecnico (
dni varchar(50)
constraint tecnico_pk primary key (dni)
);
```

```
create table Operador (
dni varchar(50)
constraint operador_pk primary key (dni)
)
```

```
create table Proyecto (
num_proyecto varchar(15),
nombre varchar(50),
num_tipo varchar (5),
dni_tecnico varchar(50) not null,
constraint proyecto_pk primary key (num_proyecto),
constraint proyecto_fk foreign key (num_tipo) references Tipo_Proyecto(num_tipo)
);
```

```
create table tipoproyecto(
num_tipo varchar(5),
nombre varchar(50),
constraint tipoproyecto_pk primary key (num_tipo)
);
```

```
create table Fase (
num_fase varchar(4) ,
nombre varchar (50) ,
num_proyecto varchar(15),
constraint fase_pk primary key (num_fase, num_proyecto),
constraint fase_fk foreign key (num_proyecto) references Proyecto(num_proyecto)
)
```

### **3.2 Inserción de datos**

```
insert into empleado value ('123', 'juan', 35, NULL), ('456', 'jose', 45, '123'), ('789', 'andres', 41, '123'), ('012', 'pilar',
67, '456');
insert into telefono ('1234567'), ('8912345');
insert into telefonoempleado('123','1234567'), ('456','1234567');
insert into tecnico ( '123'), ( '456');
```

```
insert into operador ( '789'), ('012');  
insert into proyecto value ( 'p1', 'casa', 't1','123'), ( 'p2', 'puente', 't2','456');  
insert into tipoproyecto value ('t1','viviendas') ,('t2','carreteras');  
insert into fase value ('f1','cimientos','p1'),('f2','pared','p1');  
insert into trabaja value ('p1','f1','789'), ('p1','f1','012');
```

### **3.2 Borrado de tablas**

*drop table trabaja, fase, proyecto, tipoproyecto, operador, tecnico, telefonoempleado, telefono, empleado*