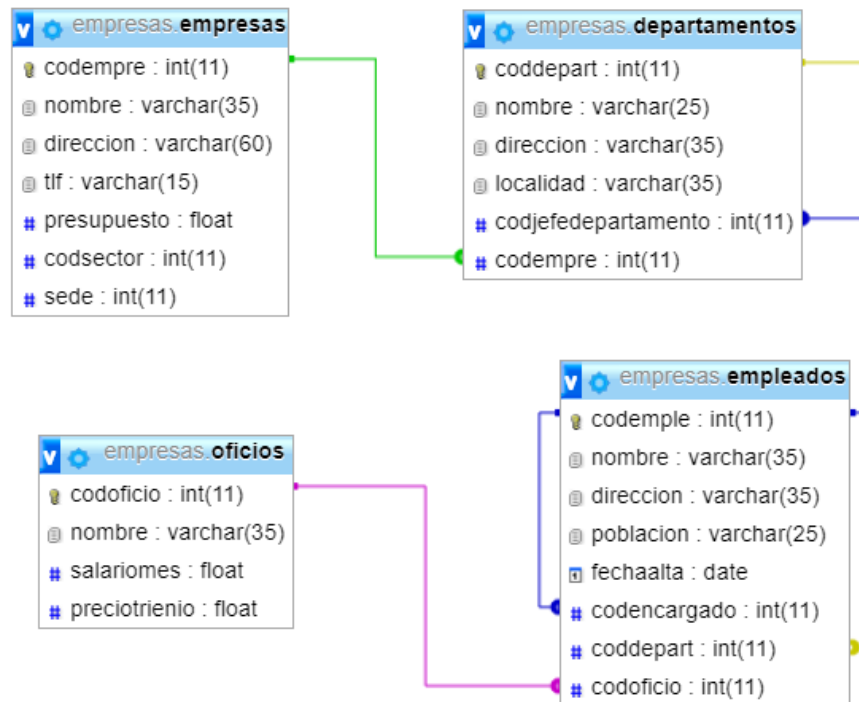


RA2 (17%)

Evaluación de los criterios:

f) 25%. Se han desarrollado aplicaciones que modifican el contenido de la base de datos. **EJERCICIO3****h) 20%.** Se han desarrollado aplicaciones que efectúan consultas. **EJERCICIO1****i) 25%.** Se han ejecutado procedimientos en la base de datos. **EJERCICIO2**

Dada la BD MySQL de nombre **empresas**, cuyo modelo de datos se presenta en la figura:



Donde las tablas contienen la información de una red de empresas con los datos de los departamentos, y empleados de las empresas. Las tablas son las siguientes:

- **EMPRESAS** – Esta tabla contiene la información de las distintas empresas. La clave es el código de empresa **codempre**. La empresa tiene la sede en un país, y además opera en muchos países. La empresa también pertenece a un sector. Del sector se pueden recibir ayudas. También la empresa está formada por departamentos, que es donde van a trabajar los empleados. Una empresa puede tener varios departamentos.
- **DEPARTAMENTOS** – Contiene la información de los distintos departamentos de la empresa. La empresa puede tener varios departamentos. Cada departamento pertenece a una empresa. La clave es **coddepart**. Un departamento además va a tener de jefe a un empleado.
- **EMPLEADOS** – Contiene la información de los empleados de las empresas. Cada empleado pertenece a un departamento. Además, cada empleado va a tener a un encargado que lo supervisa, y que será otro empleado. También el empleado estará asignado a un oficio. La clave es el código de empleado **codemple**.
- **OFICIOS** – Contiene la información de los oficios, el salario mensual y el coste de los trienios. La clave es el código de oficio **codoficio**. A un oficio pertenecen muchos empleados.

El usuario de la base de datos se llamará **empresas** con clave **empresas**. Se adjunta el script para crear las tablas de esa BD. La BD y el usuario lo puedes crear usando PhpMyAdmin. La conexión a la base de datos se hará así:

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection conexion = DriverManager.getConnection("jdbc:mysql://localhost/empresas",
"empresas", "empresas");
```

A partir de estas tablas. Crear un proyecto Eclipse de nombre **RA2_APELLIDOSNOMBRE**, dentro una clase java de nombre **Main_ApellidosNombre.java** y realiza los siguientes ejercicios (cada ejercicio es un método):

SE PIDE:

(H) EJERCICIO1: consultas

Realiza un método java de nombre **ejercicio1()** que reciba un código de empresa, y visualice todos los departamentos de la empresa con sus empleados.

Si la empresa no existe, visualizar que el código de empresa no existe en la BD y no mostrar el listado.
Y si la empresa existe se debe de visualizar este informe con la siguiente información:

```
COD-EMPRESA: xxx    NOMBRE: xxxxxxxxxxxxxxxxxxxx
DIRECCIÓN: xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx    Número de departamentos: xxx
-----
COD-DEPARTAMENTO: xxxx    NOMBRE: xxxxxxxxx    LOCALIDAD: xxxxxxxxx
-----
COD-EMP    NOMBRE    OFICIO    NOMBRE ENCARGADO
-----
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx

Número de empleados del departamento:  xx
Nombre del jefe del departamento: xxxxxxxxx

-----
COD-DEPARTAMENTO: xxxx    NOMBRE: xxxxxxxxx    LOCALIDAD: xxxxxxxxx
-----
COD-EMP    NOMBRE    OFICIO    NOMBRE ENCARGADO
-----
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx
xxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx    xxxxxxxxxxxxxxxxxxxx

Número de empleados del departamento:  xx
Nombre del jefe del departamento: xxxxxxxxx

.....
.....
```

Por ejemplo para el código de empresa 1 que tiene dos departamentos, se debe mostrar:

```
COD-EMPRESA: 1    NOMBRE: Comercios Madrid
DIRECCIÓN: Polígono 724. 28211. Madrid    Número de departamentos: 2
-----
COD-DEPARTAMENTO: 11    NOMBRE: GESTIÓN    LOCALIDAD: Madrid
-----
COD-EMP    NOMBRE    OFICIO    NOMBRE ENCARGADO
-----
101    Pedro Martínez    INGENIERO AGRÍCOLA    ***
107    Juana Salazar    TÉCNICO ADMINISTRATIVO    Pedro Martínez

Número de empleados del departamento:  2
```

Nombre del jefe del departamento: Pedro Martínez

COD-DEPARTAMENTO: 12 NOMBRE: ALMACÉN LOCALIDAD: Talavera de la Reina

COD-EMP	NOMBRE	OFICIO	NOMBRE ENCARGADO
102	Luisa Ros	INGENIERO AGRÍCOLA	Pedro Martínez
103	Juan Sales	TÉCNICO GANADERO	Luisa Ros
104	Antonio Rios	TÉCNICO GANADERO	Luisa Ros
105	Alberto Martín	PEÓN GANADERO	Luisa Ros
106	María Suarez	PEÓN GANADERO	Luisa Ros

Número de empleados del departamento: 5

Nombre del jefe del departamento: Luisa Ros

En nombre de encargado se mostrará *** cuando el codencargado sea nulo

(I) EJERCICIO2: ejecutar procedimientos de base de datos

Crea un método java de nombre **ejercicio2(parámetros)** que reciba un **código de empresa** y muestre en pantalla el número de departamentos que tiene y el número de empleados. Para calcular el número de departamentos realizarás **una función almacenada** (de nombre **FUN1_TUNOMBRE**) que reciba el código de la empresa y devuelva el número de departamentos. Igualmente para calcular el número de empleados realiza otra **función almacenada** (de nombre **FUN2_TUNOMBRE**) que reciba el código de la empresa y devuelva el número de empleados.

Ejemplo al probar las siguientes llamadas (debes incluirlas así en tu método main):

```
ejercicio2(1);
ejercicio2(1000);
ejercicio2(4);
```

Código empresa: 1, Nombre: Comercios Madrid

Número de departamentos: 2

Número de empleados: 7

La empresa: 1000, no existe.

Código empresa: 4, Nombre: Construcciones Industriales S.L.

Número de departamentos: 2

Número de empleados: 9

(F) EJERCICIO3: modificar contenido de la bd.

Realiza un método java de nombre **ejercicio3 (lista de parámetros)** para insertar registros en la tabla **DEPARTAMENTOS**. El método recibirá el código de departamento, nombre, dirección, localidad, código del jefe del departamento y el código de la empresa.

Antes de insertar en la tabla de **DEPARTAMENTOS** debes **añadir una columna de tipo numérico** a la tabla **EMPRESAS**. Controlar errores para el caso que se ejecute varias veces el método y ya exista la columna. A esa columna le sumarás 1 cada vez que se inserte un departamento en dicha empresa.

- Antes de insertar en **DEPARTAMENTOS** hay que comprobar que el código de departamento no exista, el código del jefe del departamento sí exista y el código de la empresa sí exista. Si no se cumplen las condiciones, se mostrarán los mensajes oportunos y no se insertará. Tampoco se sumará 1 a la columna creada en la tabla **EMPRESAS**.
- Se deben visualizar mensaje de todo lo que ocurra, si se inserta o si no, o si hay algún dato erróneo. Hay que visualizar por qué no se ha insertado, indicando **TODOS LOS MENSAJES DE ERROR OCURRIDOS**:
 - Código de departamento ya existe.
 - Jefe del departamento NO existe
 - Empresa NO existe

Ejemplo al probar las siguientes llamadas (debes incluirlas así en tu método main):

```
ejercicio3(11, "GESTIÓN", "C/Mayor 17", "Madrid", 101, 1);
ejercicio3(111, "GESTIÓN", "C/Mayor 17", "Madrid", 10, 123);
ejercicio3(11, "GESTIÓN", "C/Mayor 17", "Madrid", 10, 123);
ejercicio3(112, "GESTIÓN", "C/Mayor 17", "Madrid", 101, 1);
ejercicio3(113, "GESTIÓN", "C/Mayor 17", "Madrid", 101, 2);
```

Se mostrará esta salida:

```
EL DEPARTAMENTO 11, ya existe
HAY ERRORES, NO SE INSERTARÁ EL REGISTRO
```

```
EL JEFE DE DEPARTAMENTO 10, NO existe
EL CODIGO DE EMPRESA 123, NO existe
HAY ERRORES, NO SE INSERTARÁ EL REGISTRO
```

```
EL DEPARTAMENTO 11, ya existe
EL JEFE DE DEPARTAMENTO 10, NO existe
EL CODIGO DE EMPRESA 123, NO existe
HAY ERRORES, NO SE INSERTARÁ EL REGISTRO
```

```
Registro INSERTADO....
Se ha sumado 1 a la empresa: 1
```

```
Registro INSERTADO....
Se ha sumado 1 a la empresa: 2
```

Cada ejercicio se valorará con 10 puntos.

Los ejercicios se ejecutarán en el orden indicado.

FORMATO DE ENTREGA

CREA UN FICHERO COMPRIMIDO DE NOMBRE: **RA2_ApellidosNombre.zip** que contenga:

- El proyecto Java con nombre **RA2_ApellidosNombre**, dentro estará la clase con el método **main()** que lanza los 3 ejercicios. Esta clase se llamará **MainApellidosNombre.java**.
- El ejecutable **Main.jar** que lanza la aplicación.

Para probar el ejercicio se ejecutará el JAR desde la línea de comandos del DOS: **java -jar Main.jar**

Cualquier excepción no controlada que hace que se detenga la ejecución del programa resta 5 puntos del ejercicio.

Entorno de ejecución JDK 1.8