Ejercicio ORM 2: relaciones 1 a muchos

1. Crear un programa que cree dos sedes nuevas

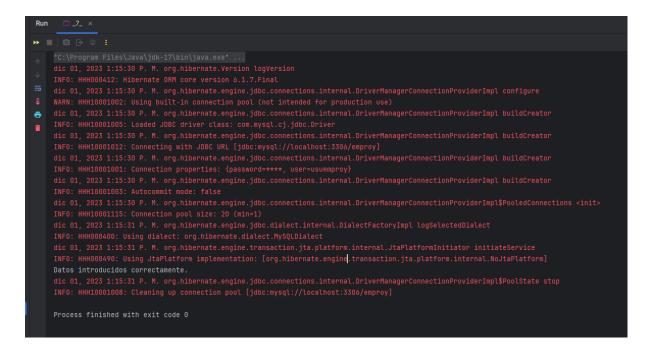
```
C:\Program Files\Java\jdk-17\bin\java.exe" ...

dic 01, 2023 1:12:55 P. M. org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 6.1.7.Final
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH0001002: Using built-in connection pool (not intended for production use)
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Loaded JBBC driver class: com.mysql.cj.jdbc.Driver
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001012: Connecting with JDBC URL [jdbc:mysql://locahlost:3306/emproy]
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {password=****, user=usuemproy}
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autcommit mode: false
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH10001105: Connection pool size: 20 (min=1)
dic 01, 2023 1:12:55 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH0004008: Using dialect: org.hibernate.engine.jdbc.connections.internal.JtaPlatformInitiator initiateService
INFO: HHH0004008: Using dialect: org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH00004008: Using dialect: org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH00004008: Using dialect: org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH00004008: Using dialect: org.hibernate.engine.jdbc.connections.internal.Driver
```

2. Crear un programa que para alguna de las sedes anteriores cree dos departamentos suyos.

3. Crear cuatro empleados y asignar dos de ellos a un departamento de los antes creados, y los otros dos a otro departamento

```
public static void main(String[] args) {
    try (SessionFactory sessionFactory = new Configuration().configure().
            buildSessionFactory();
         Session s = sessionFactory.openSession()) {
            \underline{t} = s.beginTransaction();
            Departamento marketing = s.get(Departamento.class, 0:1);
            Departamento programacion = s.get(Departamento.class, 0:2);
           Empleado e1 = new Empleado();
           e1.setDni("26524624N");
           e1.setNomEmp("Sergio");
           e1.setDepartamentoByIdDepto(programacion);
           Empleado e2 = new Empleado();
           e2.setDni("26125129C");
           e2.setNomEmp("Paco");
           e2.setDepartamentoByIdDepto(marketing);
           Empleado e3 = new Empleado();
           e3.setDni("19561802A");
           e3.setNomEmp("Isabel");
           e3.setDepartamentoByIdDepto(programacion);
           Empleado e4 = new Empleado();
           e4.setNomEmp("Maria");
            e4.setDepartamentoByIdDepto(marketing);
            s.save(e4);
            t.commit();
            System.out.println("Datos introducidos correctamente.");
        } catch (Exception ex) {
               System.out.println("ERROR: ROLLBACK");
```



4. Crear un programa que, dado un código de una sede por línea de comandos, muestre todos los departamentos de la sede, obtenidos de un atributo de clase Sede.

```
•
\blacktriangleright
          public static void main(String[] args) {
             Scanner sc = new Scanner(System.in);
              try (SessionFactory sessionFactory = new Configuration().configure().
                    buildSessionFactory();
                  Session s = sessionFactory.openSession()) {
                      t = s.beginTransaction();
                      System.out.println("Escribeme el codigo de la sede");
                      Sede sede = s.get(Sede.class, codSede);
                      t.commit();
                      Collection<Departamento> dep = sede.getDepartamentosByIdSede();
                      for (Departamento departamento : dep) {
                          System.out.println(departamento.getNomDepto());
                  } catch (Exception ex) {
                      ex.printStackTrace(System.err);
                          System.out.println("ERROR: ROLLBACK");
```