

- 1º Lugar creamos e insertamos el script que compone el modelo de datos de la BD “gestiónproyectos.sql”, el cual puede encontrar en la carpeta raíz de este proyecto.
- 2º Creamos la conexión en el ID Netbeans a la base de datos, dirigiéndonos a la pestaña services y realizando click derecho para crear una nueva conexión, donde indicaremos el nombre de la base de datos en cuestión, usuario y contraseña (Imagen 1),obteniendo como resultado la (imagen 2).

Imagen 1:

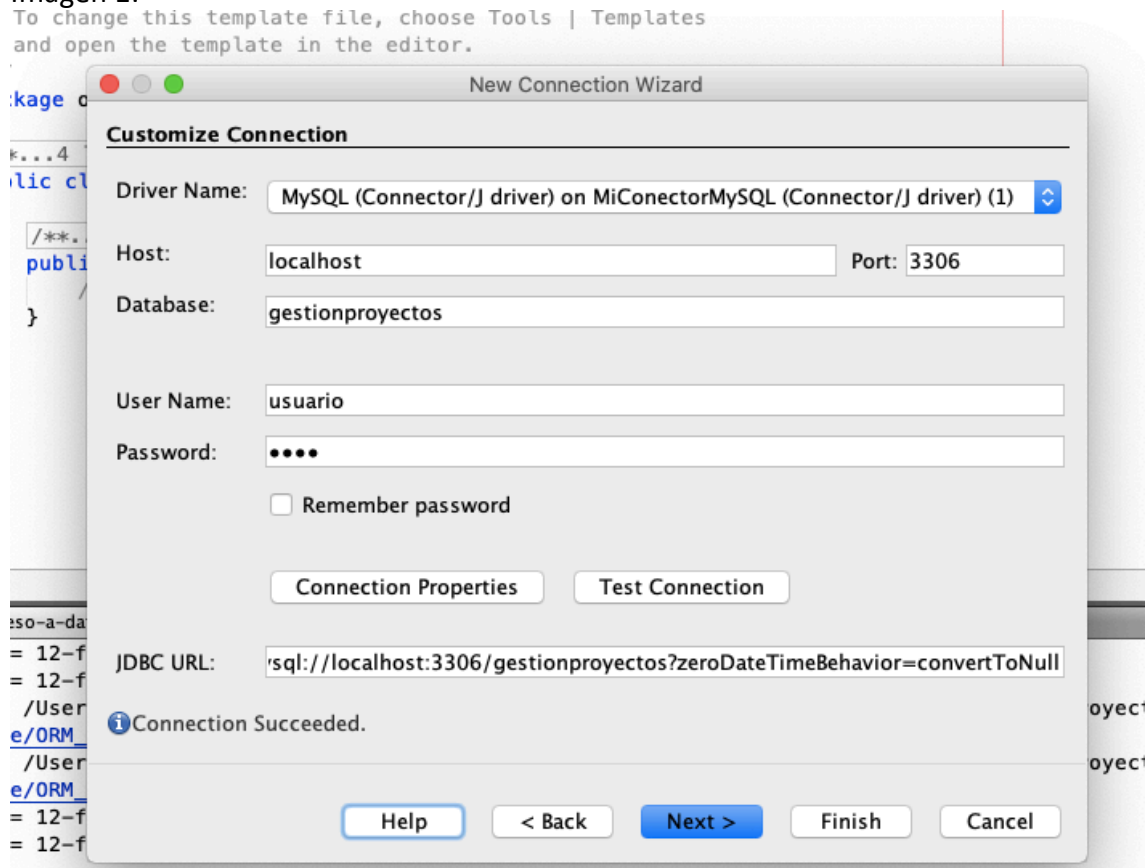
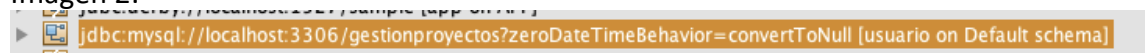


Imagen 2:



3º Crearemos los ficheros necesarios para el proyecto de Hibernate usando el asistente (Imagen 1), una vez creado todos(excepto Hibernate Mapping Wizard) nos quedara como resultado la (Imagen 2).

Imagen1:

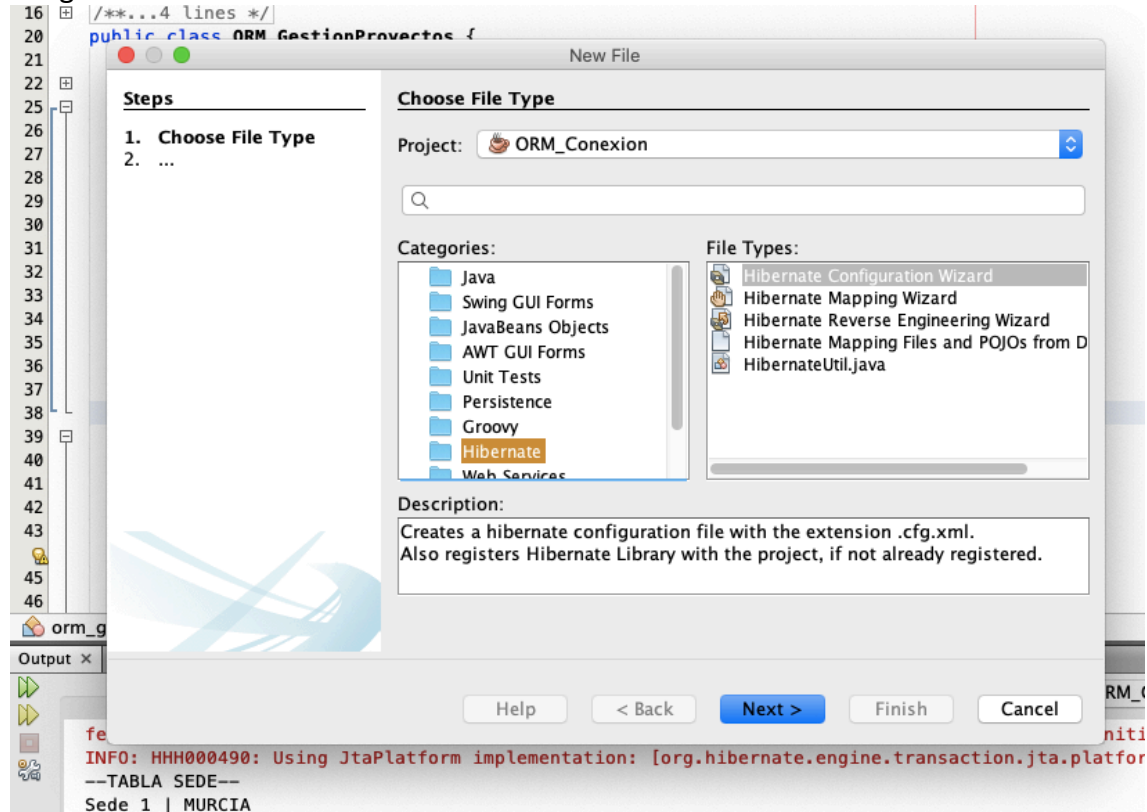
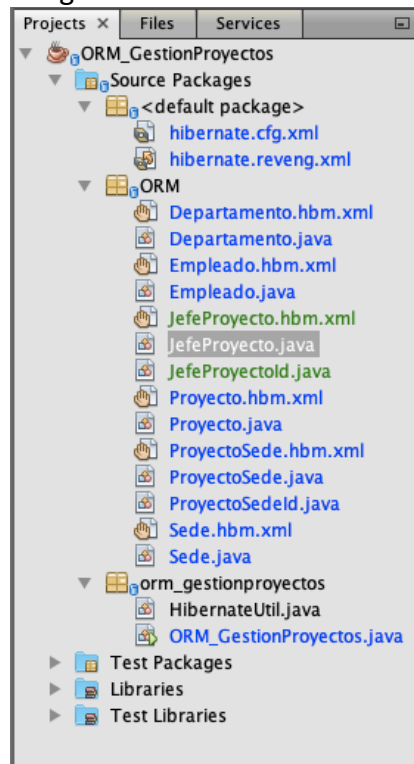


Imagen 2:



4º Comprobamos que la conexión de nuestro proyecto a la base de datos es correcta realizando una simple consulta HQL sobre una de las tablas maestras, la cual tiene algunos datos insertados de prueba (Imagen 1).

Imagen 1:

The screenshot shows an IDE with a Java file named `ORM_GestionProyectos` and its MySQL database output. The Java code includes a `main` method that uses Hibernate to open a session and execute an HQL query to select all records from the `Sede` table. The output window shows the successful execution of the query, displaying two rows of data: `Sede 1 | MURCIA` and `Sede 2 | BARCELONA`. The MySQL console shows the schema of the `sede` table and the results of the `select * from sede;` query.

```

16  /**...4 lines */
20  public class ORM_GestionProyectos {
21
22      /**...3 lines */
25  public static void main(String[] args) {
26      // TODO code application logic here
27      Transaction t = null;
28      try (Session s = HibernateUtil.getSessionFactory().openSession()) {
29          SelectsHQL(s);
30
31      } catch (Exception e) {
32          e.printStackTrace(System.err);
33          if (t != null) {
34              t.rollback();
35          }
36      }
37
38  }
39
40  public static void SelectsHQL(Session s) {
41      //Consultas Select de HQL
42      Query query = s.createQuery("FROM Sede").setResultType(Sede.class);
43      List<Sede> lSede = query.getResultList();
44      System.out.println("--TABLA SEDE--");
45      for (Sede item : lSede) {
46          System.out.println("Sede " + item.getIdSede() + " | " + item.getNomSede());
47      }
48  }
49
50  }
51
52  }
53
54  }
55
56  }
57
58  }
59
60  }
61
62  }
63
64  }
65
66  }
67
68  }
69
70  }
71
72  }
73
74  }
75
76  }
77
78  }
79
80  }
81
82  }
83
84  }
85
86  }
87
88  }
89
90  }
91
92  }
93
94  }
95
96  }
97
98  }
99
100 }

```

```

mysql> show create table sede;
+-----+
| Table | `sede` |
+-----+
| Engine | InnoDB |
| Charset | utf8mb4 |
| Collation | utf8mb4_0900_ai_ci |
+-----+
mysql> insert into sede (id_sede,nom_sede)VALUES(1,"MURCIA");
Query OK, 1 row affected (0,00 sec)
mysql> insert into sede (id_sede,nom_sede)VALUES(2,"BARCELONA");
Query OK, 1 row affected (0,00 sec)
mysql> select * from sede;
+-----+
| id_sede | nom_sede |
+-----+
| 1 | MURCIA |
| 2 | BARCELONA |
+-----+
2 rows in set (0,00 sec)
mysql>

```

5º Creamos alguno empleado y proyecto, y asignamos a este como jefe del proyecto (Imagen 1).

The screenshot shows an IDE with a Java file named `JefeProyectoId` and its MySQL database output. The Java code includes a `main` method that creates a `Departamento`, a `Proyecto`, an `Empleado`, and a `JefeProyectoId` object. It then assigns the employee as the project manager and saves all objects. The output window shows the successful execution of the code, displaying the details of the created objects. The MySQL console shows the schema of the `empleados`, `proyectos`, and `departamentos` tables and the results of the `select * from empleados;`, `select * from proyectos;`, and `select * from departamentos;` queries.

```

1  s.save(s1);
2  //Creamos el departamento
3  Departamento d1 = new Departamento();
4  d1.setIdDepto(1);
5  d1.setNomDepto("INFORMATICA");
6  d1.setSede(s1);
7  s.save(d1);
8  //Creamos el proyecto
9  Proyecto p1 = new Proyecto();
10 p1.setIdProy(1);
11 p1.setNomProy("ODISEA");
12 p1.setFinicio(new Date());
13 s.save(p1);
14 //Creamos el empleado
15 Empleado e1= new Empleado();
16 e1.setDni("12345678L");
17 e1.setNomEmp("PABLO");
18 e1.setDepartamento(d1);
19 s.save(e1);
20 //Asignamos un empleado a un proyecto como jefe, e indicamos des
21 JefeProyectoId jefeProyectoId = new JefeProyectoId(); //Esta cla
22 jefeProyectoId.setDni(e1.getDni());
23 jefeProyectoId.setIdProy(p1.getIdProy());
24 JefeProyecto j = new JefeProyecto(); //Esta otra clase es para el
25 j.setId(jefeProyectoId);
26
27 j.setEmpleado(e1);
28 j.setProyecto(p1);
29 j.setFinicio(new Date());
30 s.save(j);
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

```

mysql> select * from empleados;
+-----+
| dni | nom_emp | id_depto |
+-----+
| 12345678L | PABLO | 2 |
+-----+
1 row in set (0,00 sec)
mysql> select * from jefe_proyecto;
+-----+
| dni | id_proy | f_inicio | f_fin |
+-----+
| 12345678L | 2 | 2020-02-13 | NULL |
+-----+
1 row in set (0,00 sec)
mysql> select * from proyecto;
+-----+
| id_proy | f_inicio | f_fin | nom_proy |
+-----+
| 2 | 2020-02-13 | NULL | ODISEA |
+-----+
1 row in set (0,00 sec)
mysql> select * from departamento;
+-----+
| id_depto | nom_depto | id_sede |
+-----+
| 2 | INFORMATICA | 2 |
+-----+
1 row in set (0,00 sec)
mysql> select * from sede;
+-----+
| id_sede | nom_sede |
+-----+
| 2 | GRANADA |
+-----+
1 row in set (0,00 sec)
mysql>

```