



Acceso a Datos

Tarea 9.1: Proyecto Java +
ObjectDB



1. Código de entidades

- *Persona*

```
objectDBLuciana > objectdbluciana > src > main > java > com > example > Person.java
1 package com.example;
2
3 import javax.persistence.Entity;
4 import javax.persistence.Id;
5
6 @Entity
7 public class Persona {
8     @Id
9     private int id;
10    private String nombre;
11    private int edad;
12
13    public Persona() {
14    }
15    public Persona(int id, String nombre, int edad) {
16        this.id = id;
17        this.nombre = nombre;
18        this.edad = edad;
19    }
20}
21
```

- *Empleado*

```
pom.xml Departamento.java 1 Empleado.java 2 Main.java 1 Ciudad.java 1 Persona.java 2
objectDBLuciana > objectdbluciana > src > main > java > com > example > Empleado.java > Empleado > Empleado(int, String, int)
1
2 import javax.persistence.Entity;
3 import javax.persistence.Id;
4 import javax.persistence.ManyToOne;
5
6 @Entity
7 public class Empleado extends Persona {
8     private double salario;
9
10    @ManyToOne
11    Departamento departamento;
12
13    // Añadido
14    Ciudad ciudad;
15
16    public Empleado() {}
17
18    public Empleado(int id, String nombre, int edad, double salario, Departamento d, Ciudad c) {
19        super(id, nombre, edad);
20        this.salario = salario;
21        this.departamento = d;
22        this.ciudad = c;
23    }
24}
25
```



- *Departamento*

```
pom.xml Departamento.java 1 Empleado.java 2
objectDBLuciana > objectdbluciana > src > main > java > com > example >
1 package com.example;
2
3 import javax.persistence.Entity;
4 import javax.persistence.Id;
5
6 @Entity
7 public class Departamento {
8     @Id
9     private int id;
10    private String nombre;
11
12    public Departamento() {
13    }
14
15    public Departamento(int id, String nombre) {
16        this.id = id;
17        this.nombre = nombre;
18    }
19 }
```

Ampliación:

- *Ciudad*

```
jectDBLuciana > objectdbluciana > src > main > java > com > e
1 package com.example;
2
3 import javax.persistence.Entity;
4 import javax.persistence.Id;
5
6 @Entity
7 public class Ciudad {
8
9     @Id
10    private int id;
11    private String nombre;
12
13    public Ciudad(int id, String nombre){
14        this.id = id;
15        this.nombre = nombre;
16    }
17
18 }
19 }
```



2. Main.java

```
objectDBLuciana > objectdbluciana > src > main > java > com > example > Main.java > Main > main(String[])
6
7 public class Main {
8     Run | Debug
9     public static void main(String[] args) {
10         // Ruta directa al archivo .odb
11         EntityManagerFactory emf =
12             Persistence.createEntityManagerFactory(persistenceUnitName: "C:\\\\Users\\\\PC218\\\\Desktop\\\\objectdb-2.9.4\\\\db\\\\prueba.odb");
13
14         EntityManager em = emf.createEntityManager();
15
16         em.getTransaction().begin();
17
18         Departamento inf = new Departamento(id: 1, nombre: "Informática");
19         Departamento rrhh = new Departamento(id: 2, nombre: "RRHH");
20
21         // NUEVOS DEPARTAMENTOS
22         Departamento cont = new Departamento(id: 3, nombre: "Contabilidad");
23         Departamento inv = new Departamento(id: 4, nombre: "Investigación");
24
25         // CIUDADES
26         Ciudad c1 = new Ciudad(id: 1, nombre: "Madrid");
27         Ciudad c2 = new Ciudad(id: 2, nombre: "Valencia");
28         Ciudad c3 = new Ciudad(id: 3, nombre: "Burgos");
29         Ciudad c4 = new Ciudad(id: 4, nombre: "Sevilla");
30         Ciudad c5 = new Ciudad(id: 5, nombre: "Almería");
31
32         // AÑADIMOS LAS CIUDADES
33         Empleado e1 = new Empleado(id: 1, nombre: "Ana", edad: 30, salario: 1500, inf, c1);
34         Empleado e2 = new Empleado(id: 2, nombre: "Luis", edad: 45, salario: 2000, inf, c2);
35         Empleado e3 = new Empleado(id: 3, nombre: "Marta", edad: 28, salario: 1300, rrhh, c2);
36         Empleado e4 = new Empleado(id: 4, nombre: "Pedro", edad: 32, salario: 1200, rrhh, c5);
37
38         // NUEVOS EMPLEADOS
39         Empleado e5 = new Empleado(id: 5, nombre: "Martín", edad: 24, salario: 2100, cont, c2);
40         Empleado e6 = new Empleado(id: 6, nombre: "Sara", edad: 54, salario: 1200, inv, c4);
41         Empleado e7 = new Empleado(id: 7, nombre: "Esperanza", edad: 28, salario: 1200, inv, c2);
42         Empleado e8 = new Empleado(id: 8, nombre: "Francisco", edad: 46, salario: 2000, cont, c3);
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
```

Ready

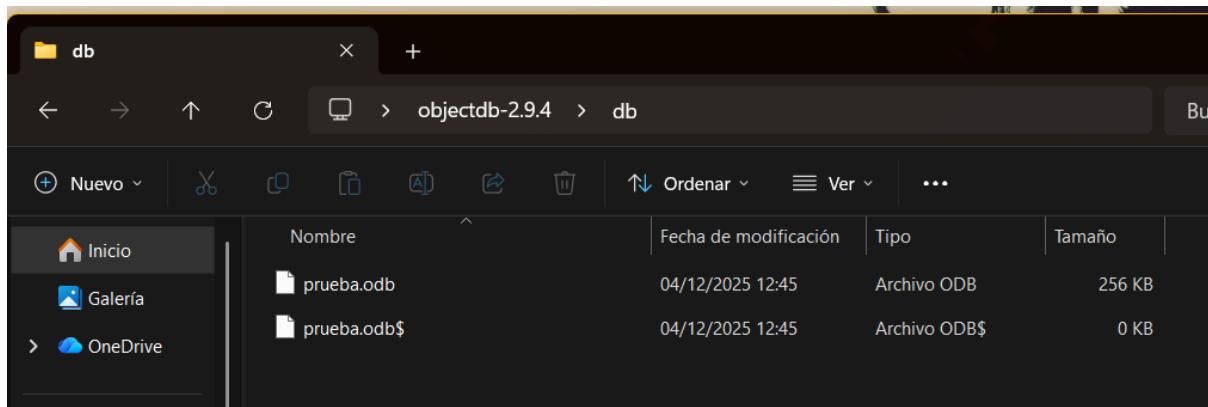
Ln 29, Col 46 Spaces: 4 UTF-8 LF {} Java Signed out Go Live

```
pom.xml Departamento.java 1 Empleado.java 2 Main.java 1 Ciudad.java 1 Personaje.java 2
objectDBLuciana > objectdbluciana > src > main > java > com > example > Main.java > Main > main(String[])
7 public class Main {
8     public static void main(String[] args) {
9
10         // PERSONA
11         Persona p1 = new Persona(id: 1, nombre: "Florencia", edad: 43);
12
13         // PERSISTENTES DE CIUDAD
14         em.persist(c1);
15         em.persist(c2);
16         em.persist(c3);
17         em.persist(c4);
18         em.persist(c5);
19
20         em.persist(inf);
21         em.persist(rrhh);
22
23         // PERSISTENTES DEPARTAMENTOS AÑADIDOS
24         em.persist(cont);
25         em.persist(inv);
26
27         em.persist(e1);
28         em.persist(e2);
29         em.persist(e3);
30         em.persist(e4);
31
32         // PERSISTENTES EMPLEADOS AÑADIDOS
33         em.persist(e5);
34         em.persist(e6);
35         em.persist(e7);
36         em.persist(e8);
37
38         em.getTransaction().commit();
39
40         em.close();
41         emf.close();
42
43         System.out.println("Base de datos creada con éxito.");
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
```

Ready

Ln 29, Col 46 Spaces: 4 UTF-8 LF {} Java Signed out Go Live

3. Archivo prueba.odb



4. Consultas OQL

- Consulta 1

SELECT e FROM Empleado e

Query Res...	Results	[size = 8]
0	Empleado#1	0
1	Empleado#2	0
2	Empleado#3	0
3	Empleado#4	0
4	Empleado#5	0
5	Empleado#6	0
6	Empleado#7	0
7	Empleado#8	0

- Consulta 2

SELECT e.nombre FROM Empleado e

Query Result List	Results	[size = 8]
0	String	'Ana'
1	String	'Luis'
2	String	'Marta'
3	String	'Pedro'
4	String	'Martin'
5	String	'Sara'
6	String	'Esperanza'
7	String	'Francisco'



- Consulta 3

SELECT e FROM Empleado e WHERE e.departamento.nombre = "Informática"

The screenshot shows the ObjectDB Explorer interface. On the left, the 'Query Execution' panel displays the JPQL query: 'SELECT e FROM Empleado e WHERE e.departamento.nombre = "Informática"'. On the right, the 'Query Result List' panel shows the results:

Results	[size = 2]
[0] Empleado#1	0
[1] Empleado#2	0

- Consulta 4

SELECT e FROM Empleado e WHERE e.salario > 1500

The screenshot shows the ObjectDB Explorer interface. On the left, the 'Query Execution' panel displays the JPQL query: 'SELECT e FROM Empleado e WHERE e.salario > 1500'. On the right, the 'Query Result List' panel shows the results:

Results	[size = 3]
[0] Empleado#2	0
[1] Empleado#5	0
[2] Empleado#8	0

- Consulta 5

SELECT e.nombre, e.departamento.nombre FROM Empleado e

The screenshot shows the ObjectDB Explorer interface. On the left, the 'Query Execution' panel displays the JPQL query: 'SELECT e.nombre, e.departamento.nombre FROM Empleado e'. On the right, the 'Query Result List' panel shows the results:

Results	[size = 8]
[0] Result	[Ana, Informática]
[1] Result	[Luis, Informática]
[2] Result	[Marta, RRHH]
[3] Result	[Pedro, RRHH]
[4] Result	[Martin, Contabilidad]
[5] Result	[Sara, Investigación]
[6] Result	[Esperanza, Investigación]
[7] Result	[Francisco, Contabilidad]



- Consulta 6

SELECT COUNT(e) FROM Empleado e

The screenshot shows the ObjectDB Explorer interface. The 'Query Text (JPQL or JDOQL)' field contains the query: 'SELECT COUNT(e) FROM Empleado e'. The 'Results' pane shows a single row with value 8.

5. Nuevos datos

- Consulta 1

SELECT e FROM Empleado e WHERE e.ciudad.nombre = "Valencia"

The screenshot shows the ObjectDB Explorer interface. The 'Query Text (JPQL or JDOQL)' field contains the query: 'SELECT e FROM Empleado e WHERE e.ciudad.nombre = "Valencia"'. The 'Results' pane displays four employee records, each associated with a specific department. The Log pane shows the execution plan and the SQL generated by the system.

Employee Record	Department	Employee Details
[0] Empleado#2	Departamento#1	edad: 45, id: 2, salario: 2000.0, nombre: Luis, ciudad: Ciudad#2 (id: 2, nombre: Valencia)
[1] Empleado#3	Departamento#1	edad: 28, id: 3, salario: 1300.0, nombre: Marta, ciudad: Ciudad#2 (id: 2, nombre: Valencia)
[2] Empleado#5	Departamento#2	edad: 24, id: 5, salario: 2100.0, nombre: Martin, ciudad: Ciudad#2 (id: 2, nombre: Valencia)
[3] Empleado#7	Departamento#3	edad: 28, id: 7, salario: 1200.0, nombre: Esperanza, ciudad: Ciudad#2 (id: 2, nombre: Valencia)



- Consulta 2

SELECT e.nombre, e.ciudad.nombre FROM Empleado e

The screenshot shows the ObjectDB Explorer interface. On the left, the 'Query Execution' panel contains the query text: 'SELECT e.nombre, e.ciudad.nombre FROM Empleado e'. Below it are sections for 'Parameters' (empty), 'First' (0), 'Max' (1000000), 'Reset', 'Disable Cache', and 'Execute' buttons. On the right, the results are displayed in a table titled 'SELECT e.nombre, e.ciudad.nombre FROM Empleado e'. The table has two columns: 'Query Res...' and 'Results'. The results are:

Query Res...	Results
[0]	Result [Ana, Madrid]
[1]	Result [Luis, Valencia]
[2]	Result [Marta, Valencia]
[3]	Result [Pedro, Almería]
[4]	Result [Martin, Valencia]
[5]	Result [Sara, Sevilla]
[6]	Result [Esperanza, Valencia]
[7]	Result [Francisco, Burgos]

- Consulta 3

SELECT e.ciudad.nombre, COUNT(e) FROM Empleado e GROUP BY e.ciudad

The screenshot shows the ObjectDB Explorer interface. On the left, the 'Query Execution' panel contains the query text: 'SELECT e.ciudad.nombre, COUNT(e) FROM Empleado e GROUP BY e.ciudad'. Below it are sections for 'Parameters' (empty), 'First' (0), 'Max' (1000000), and 'Execute' buttons. On the right, the results are displayed in a table titled 'SELECT e.ciudad.nombre, COUNT(e) FROM Empleado e GROUP BY e.ciudad'. The table has two columns: 'Query Result List' and 'Results'. The results are:

Query Result List	Results
[0]	Result [Almería, 1]
[1]	Result [Madrid, 1]
[2]	Result [Valencia, 4]
[3]	Result [Burgos, 1]
[4]	Result [Sevilla, 1]



6. Datos en el Explorer

