

## GUIA II – JPA

En esta guía ejercitamos la relación **UNO A UNO**

### 1) En MySQL crear la table “empleados”

```
CREATE TABLE `indra2022`.`empleados` ( `id` INT NOT NULL, `usuario` VARCHAR(45) NULL, `password` VARCHAR(45) NULL, PRIMARY KEY (`id`));
```

### Y luego insertar datos

```
INSERT INTO `indra2022`.`empleados` (`id`, `usuario`, `password`) VALUES ('1', 'ale', '1234');
```

### 2) Crear la clase Empleado en el package model:

```
package model;

import java.io.Serializable;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.OneToOne;
import javax.persistence.Table;

@Entity
@Table(name="empleados")
public class Empleado implements Serializable{
    private static final long serialVersionUID=-1L;

    @Id
    @Column(name="id")
    @GeneratedValue(strategy=GenerationType.AUTO)
    private long id;

    @Column(name="usuario")
    private String usuario;

    @Column(name="password")
    private String password;

    @OneToOne
```

```

@JoinColumn(name="id")
private Persona person;

public long getId() {
    return id;
}

public void setId(long id) {
    this.id = id;
}

public String getUsuario() {
    return usuario;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public String getPassword() {
    return password;
}

public void setPassword(String password) {
    this.password = password;
}

public Persona getPerson() {
    return person;
}

public void setPerson(Persona person) {
    this.person = person;
}
}

```

3) En el archivo persistence.xml, agregar a la clase Empleado

```

<?xml version="1.0" encoding="UTF-8"?>
<persistence version="1.0"
xmlns="http://java.sun.com/xml/ns/persistence"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
http://java.sun.com/xml/ns/persistence/persistence_1_0.xsd">
<persistence-unit name="JPA_Demo" transaction-type="RESOURCE_LOCAL">
<provider>org.hibernate.jpa.HibernatePersistenceProvider</provider>
<class>model.Persona</class>
<class>model.Empleado</class>
<properties>
<property name="hibernate.connection.username" value="root"/>
<property name="hibernate.connection.password" value="1234"/>
<property name="hibernate.connection.driver_class"
value="com.mysql.jdbc.Driver"/>
<property name="hibernate.connection.url"
value="jdbc:mysql://localhost:3306/indra2022"/>
<property name="hibernate.cache.provider_class"
value="org.hibernate.cache.NoCacheProvider"/>
</properties>
</persistence-unit>
</persistence>

```

- 4) Crear la clase DaoEmpleado en el package dao (esta clase tendrá el comportamiento de ABM)

```

5) package dao;
6)
7) import javax.persistence.EntityManager;
8) import javax.persistence.EntityTransaction;
9)
10) import model.Empleado;
11) import utils.JpaUtil;
12)
13) public class DaoEmpleado {
14)     public static Empleado find(Long id) {
15)         EntityManager em =
            JpaUtil.getEntityManagerFactory().createEntityManager();
16)         Empleado empleado = null;
17)         try {
18)             empleado = em.find(Empleado.class, id);
19)         } catch (Exception ex) {
20)             System.out.println("upss!! ha ocurrido un error");
21)             ex.printStackTrace();
22)         } finally {
23)             em.close();
24)         }
25)         return empleado;
26)     }

```

```
27)
28)     public static void create(Empleado empleado) {
29)         EntityManager em =
JpaUtil.getEntityManagerFactory().createEntityManager();
30)         em.getTransaction().begin();
31)         try {
32)             em.persist(empleado);
33)             em.getTransaction().commit();
34)         } catch (Exception ex) {
35)             em.getTransaction().rollback();
36)             System.out.println("Ha ocurrido un error al guardar");
37)         } finally {
38)             em.close();
39)         }
40)     }
41)
42)     public static void update(Empleado empleado) {
43)         EntityManager em =
JpaUtil.getEntityManagerFactory().createEntityManager();
44)         EntityTransaction tx = em.getTransaction();
45)         tx.begin();
46)         try {
47)             em.merge(empleado);
48)             tx.commit();
49)             System.out.println("Actualizacion exitosa");
50)         } catch (Exception ex) {
51)             tx.rollback();
52)             System.out.println("Ha ocurrido un erro al actualizar");
53)             ex.printStackTrace();
54)         } finally {
55)             em.close();
56)         }
57)     }
58)
59)     public static void delete(Long id) {
60)         EntityManager em =
JpaUtil.getEntityManagerFactory().createEntityManager();
61)         em.getTransaction().begin();
62)         try {
63)             Empleado p = DaoEmpleado.find(id);
64)             em.remove(p);
65)             em.getTransaction().commit();
66)
67)         } catch (Exception ex) {
68)             ex.printStackTrace();
```

```

69)         } finally {
70)             em.close();
71)         }
72)     }
73)
74) }
75)

```

5) Agregar a la clase Inicio en la rutina main, el siguiente código, para probar esta nueva clase

```

package ejecutable;

import dao.DaoEmpleado;
import dao.DaoPersona;
import model.Persona;
import model.Empleado;

public class Inicio {

    public static void main(String[] args) {
        Persona p=DaoPersona.find(3L);

        System.out.println("-----PERSONA de base de datos-----");
        System.out.println(p);
        System.out.println("-----Cambios-----");

        p.setNombre("Nadia");
        p.setApellido("Alvarez");

        DaoPersona.update(p);

        Empleado e =DaoEmpleado.find(1L);
        System.out.println("-----EMPLEADO de base de datos-----");
        System.out.println(e.getUsuario());
        System.out.println(e.getPerson().getNombre());
    }
}

```