

CURSO DE HIBERNATE Y JPA

EJERCICIO

CICLO DE VIDA EN HIBERNATE/JPA - JPA



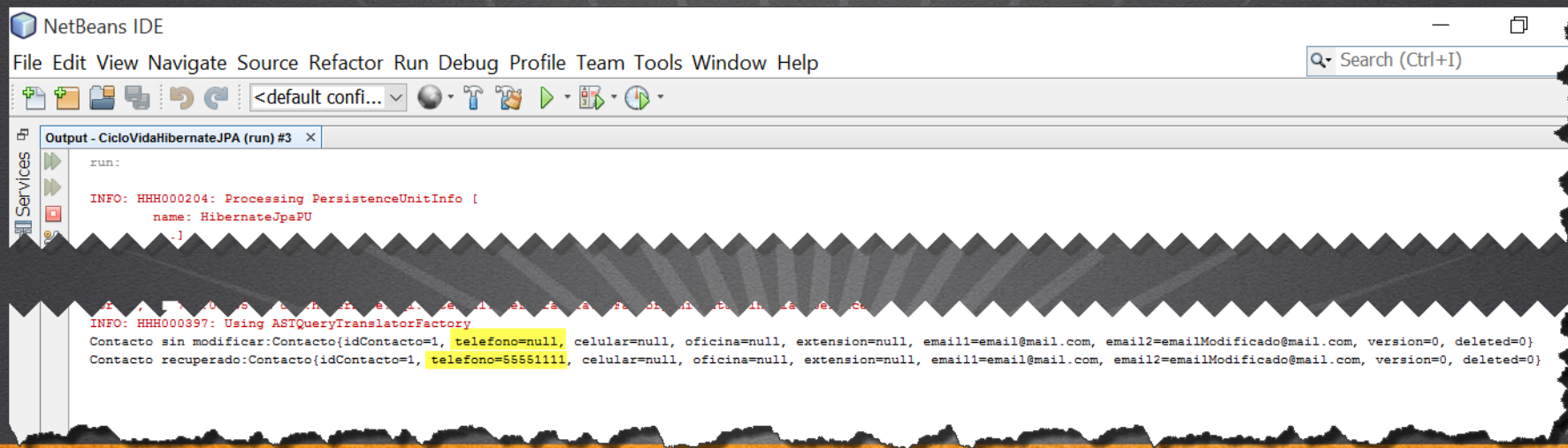
Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

OBJETIVO DEL EJERCICIO

La diferencia con este ejercicio es que no usaremos el objeto session de Hibernate, sino el objeto EntityManager de JPA. Al finalizar veremos:



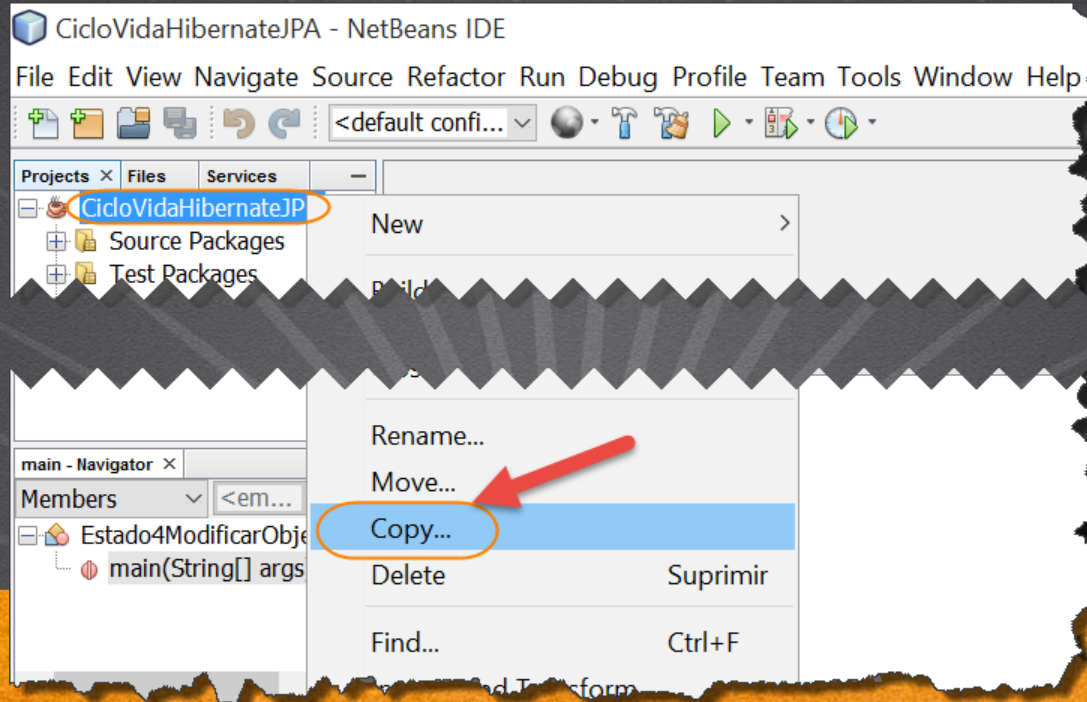
```
NetBeans IDE
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Search (Ctrl+I)
<default confi...
Output - CicloVidaHibernateJPA (run) #3
run:
INFO: HHH000204: Processing PersistenceUnitInfo [
    name: HibernateJpaPU
    ...]
INFO: HHH000397: Using ASTQueryTranslatorFactory
Contacto sin modificar:Contacto{idContacto=1, telefono=null, celular=null, oficina=null, extension=null, email1=email@mail.com, email2=emailModificado@mail.com, version=0, deleted=0}
Contacto recuperado:Contacto{idContacto=1, telefono=55551111, celular=null, oficina=null, extension=null, email1=email@mail.com, email2=emailModificado@mail.com, version=0, deleted=0}
```

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

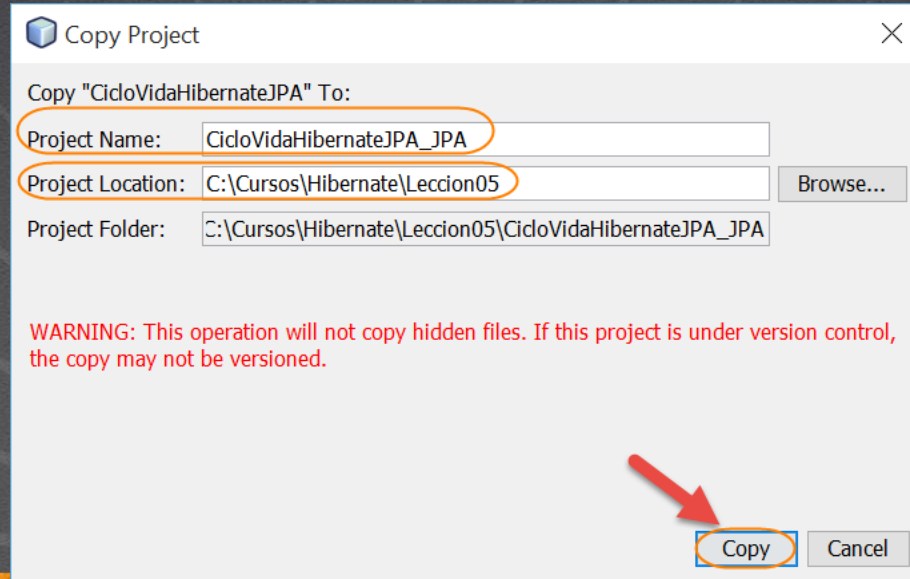
PASO 1. CREAMOS EL PROYECTO

Copiamos y pegamos el proyecto anterior
CicloVidaHibernateJPA:



PASO 1. CREAMOS EL PROYECTO

Copiamos y pegamos el proyecto anterior
CicloVidaHibernateJPA:

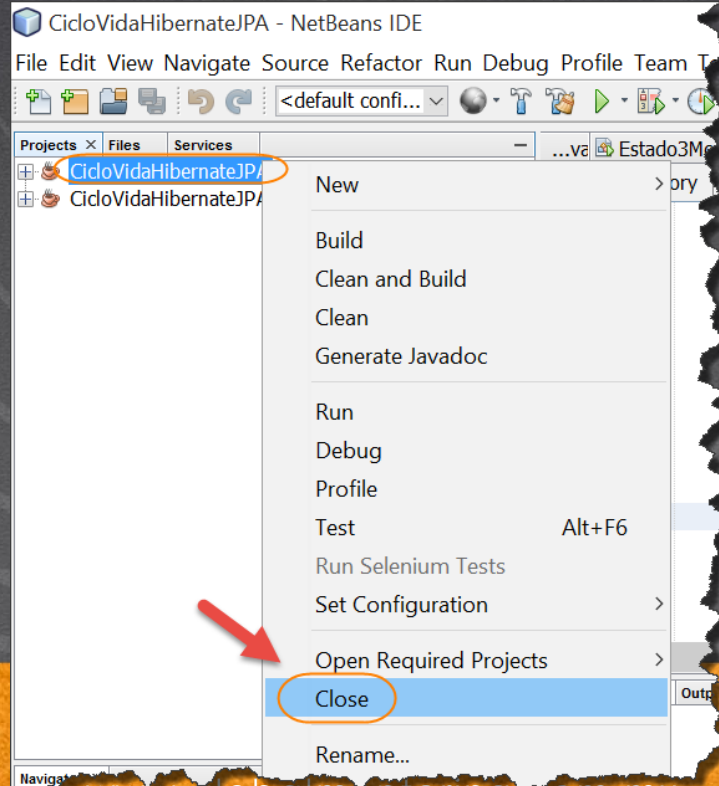


CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 2. CERRAMOS EL PROYECTO

Cerramos el proyecto que ya no utilizamos:



PASO 3. MODIFICAMOS EL CÓDIGO

[Archivo Estado1Persistido.java:](#)

Dar click para ir al código

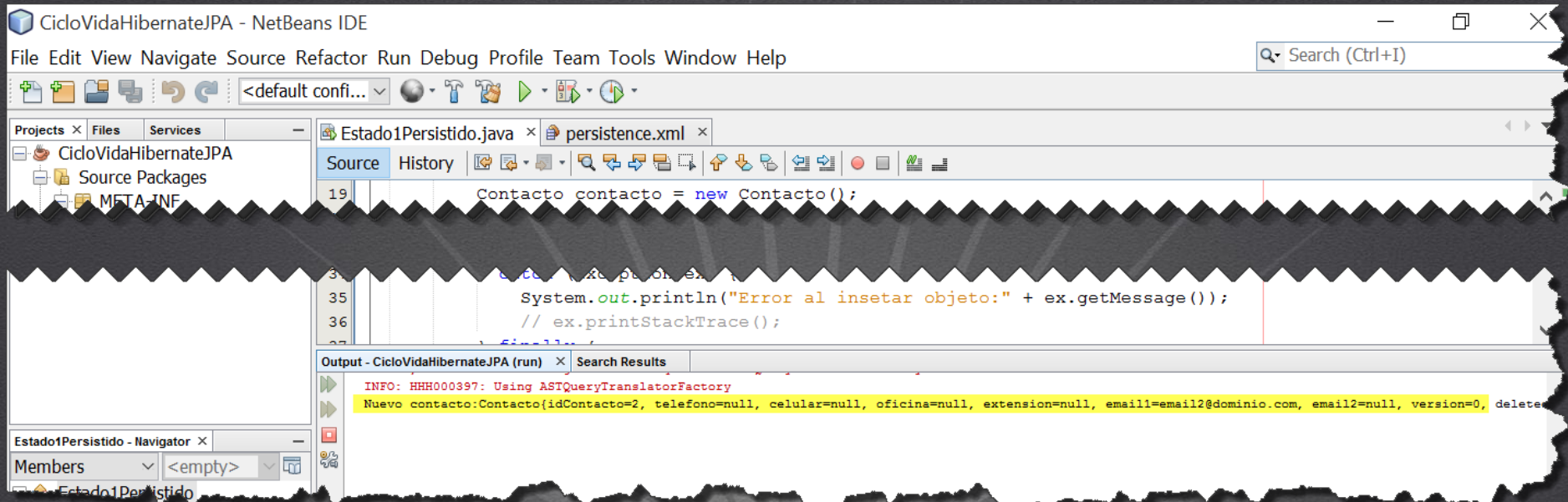
```
public static void main(String[] args) {
    /*Utilizamos la Unidad de Persistencia de JPA*/
    EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
    EntityManager em = fabrica.createEntityManager();
    /**
     * Objetivo: Estado transitorio a persistente
     */

    //1. Creamos el objeto (estado new o transitorio)
    Contacto contacto = new Contacto();
    contacto.setEmail1("email2@dominio.com");
    contacto.setVersion(0);
    contacto.setDeleted(0);

    //2. Persistimos el objeto (cambia de estado a persistente)
    //Iniciamos la transaccion
    //No utilizaremos las clases DAO en estos ejercicios
    try {
        // Iniciamos una transaccion
        em.getTransaction().begin();
        // Insertamos la nueva persona
        em.persist(contacto);
        // Terminamos la transaccion
        em.getTransaction().commit();

        //Dar click para descargar el código completo
    }
```

PASO 4. EJECUTAMOS EL CÓDIGO



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 5. MODIFICAMOS EL CÓDIGO

[Archivo Estado2RecuperarObjetoPersistente.java:](#)

Dar click para ir al código

```
public class Estado2RecuperarObjetoPersistente {

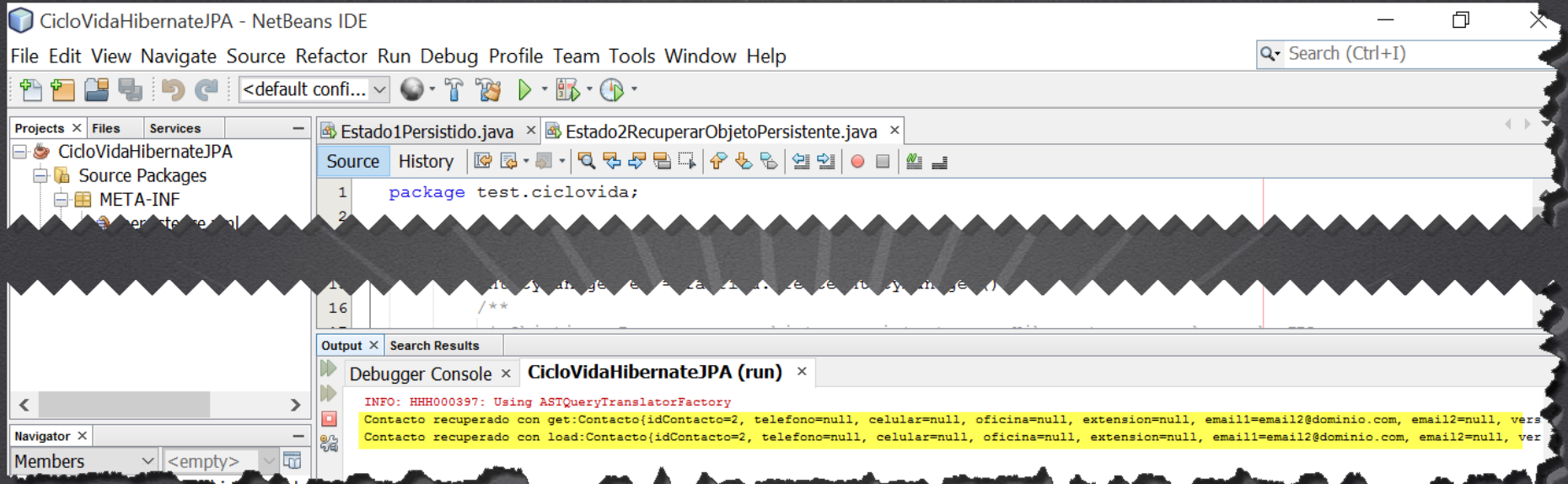
    public static void main(String[] args) {
        /*Utilizamos la Unidad de Persistencia de JPA*/
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();
        /**
         * Objetivo: Recuperar un objeto persistente con Hibernate apoyandonos de JPA
         */

        //Existen 2 maneras de realizar esta accion
        //1. Metodo get, regresa null si no encuentra el objeto
        Contacto contacto1 = null;

        try {
            em.getTransaction().begin();
            //Proporcionamos el ID a recuperar, en JPA se usa find en lugar de get
            contacto1 = (Contacto) em.find(Contacto.class, 2);
            System.out.println("Contacto recuperado con find:" + contacto1);
            em.getTransaction().commit(); //hacemos flush
        } catch (Exception e) {
            em.getTransaction().rollback();
            e.printStackTrace();
        }

        //Dar click para descargar el código completo
    }
}
```


PASO 6. EJECUTAMOS EL CÓDIGO



PASO 7. MODIFICAMOS EL CÓDIGO

[Archivo Estado3ModificarObjetoPersistente.java:](#)

Dar click para ir al código

```
public class Estado3ModificarObjetoPersistente {

    public static void main(String[] args) {
        /*Utilizamos la Unidad de Persistencia de JPA*/
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();

        /**
         * Objetivo: Modificar un objeto persistente (reattaching) Pasar de un
         * estado detached a persistente
         */

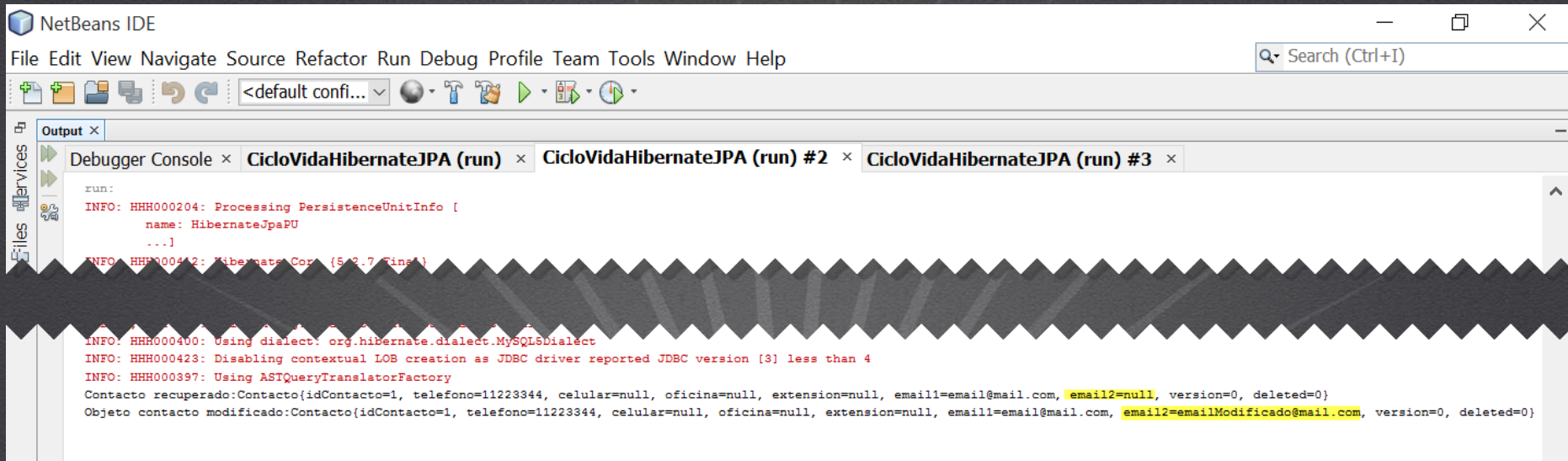
        //Recuperamos un objeto persistente
        Contacto contacto = null;
        try {
            em.getTransaction().begin();
            //Identificador a recuperar, también se puede usar merge para recuperar un objeto
            contacto = (Contacto) em.find(Contacto.class, 1);
            em.getTransaction().commit(); //hacemos flush
        } catch (Exception e) {
            em.getTransaction().rollback();
            e.printStackTrace();
        }

        //Imprimimos los valores del objeto recuperado
        //estado detached
        System.out.println("Contacto recuperado:" + contacto);

        //Modificamos el objeto fuera de una transaccion
        contacto.setEmail2("emailModificado@mail.com");

        //Dar click para descargar el código completo
    }
}
```

PASO 8. EJECUTAMOS EL CÓDIGO



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

PASO 9. MODIFICAMOS EL CÓDIGO

[Archivo Estado4ModificarObjetoSesionLarga.java:](#)

Dar click para ir al código

```
public class Estado4ModificarObjetoSesionLarga {

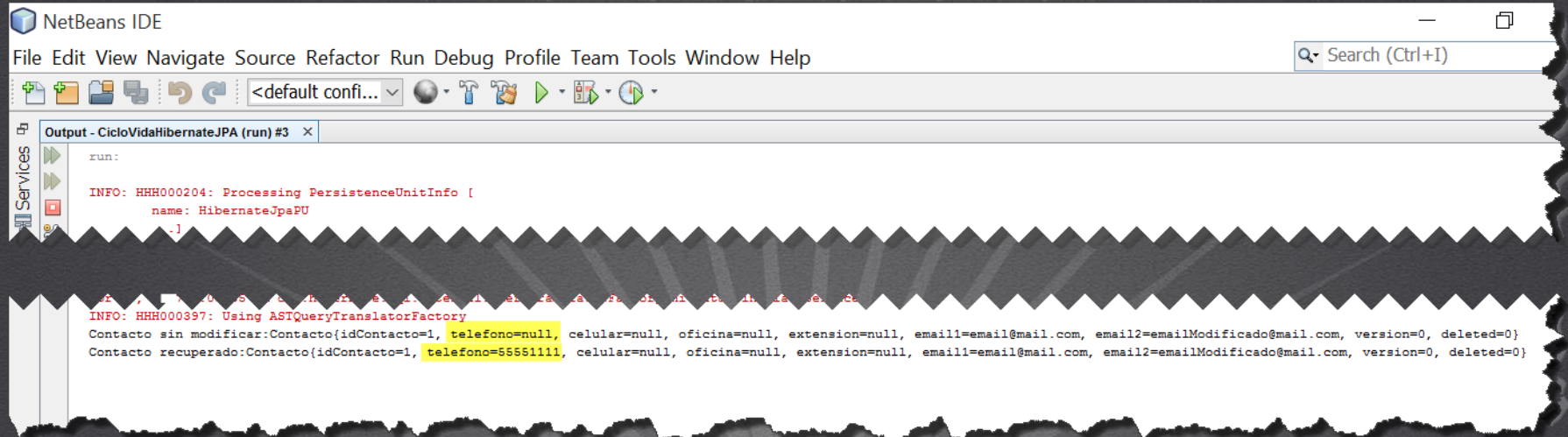
    public static void main(String[] args) {
        /*Utilizamos la Unidad de Persistencia de JPA*/
        EntityManagerFactory fabrica = Persistence.createEntityManagerFactory("HibernateJpaPU");
        EntityManager em = fabrica.createEntityManager();

        //Recuperamos un objeto persistente
        Contacto contacto = null;

        try {
            em.getTransaction().begin();
            //Identificador a recuperar, también se puede usar merge para recuperar un objeto
            contacto = (Contacto) em.find(Contacto.class, 1);
            System.out.println("Contacto sin modificar:" + contacto);
            //Modificamos el objeto en la misma transaccion
            contacto.setTelefono("55551111");
            em.getTransaction().commit(); //hacemos flush
        } catch (Exception e) {
            em.getTransaction().rollback();
            e.printStackTrace();
        } finally {
            if (em != null) {
                em.close();
            }
        }

        //Dar click para descargar el código completo
    }
}
```

PASO 10. EJECUTAMOS EL CÓDIGO



CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx

CONCLUSIÓN DEL EJERCICIO

- Con este ejercicio hemos visto el ciclo de vida de los objetos de entidad pero apoyándonos más con el API de JPA, sin utilizar el objeto session de Hibernate.
- Lo más común es que utilicemos ya el API de JPA, sin embargo hay ocasiones es que hay métodos que queremos utilizar de Hibernate, y por lo tanto nos interesa obtener el objeto session en lugar de utilizar el objeto entityManager de JPA. Así que dependerá de nuestros requerimientos usar una u otra solución.

CURSO ONLINE

HIBERNATE & JPA

Por: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE HIBERNATE Y JPA

www.globalmentoring.com.mx