

CURSO DE JAVA EE

CICLO DE VIDA EN JPA



Ing. Ubaldo Acosta

Por el experto: Ing. Ubaldo Acosta



Experiencia y Conocimiento para tu vida

CURSO DE JAVA EE

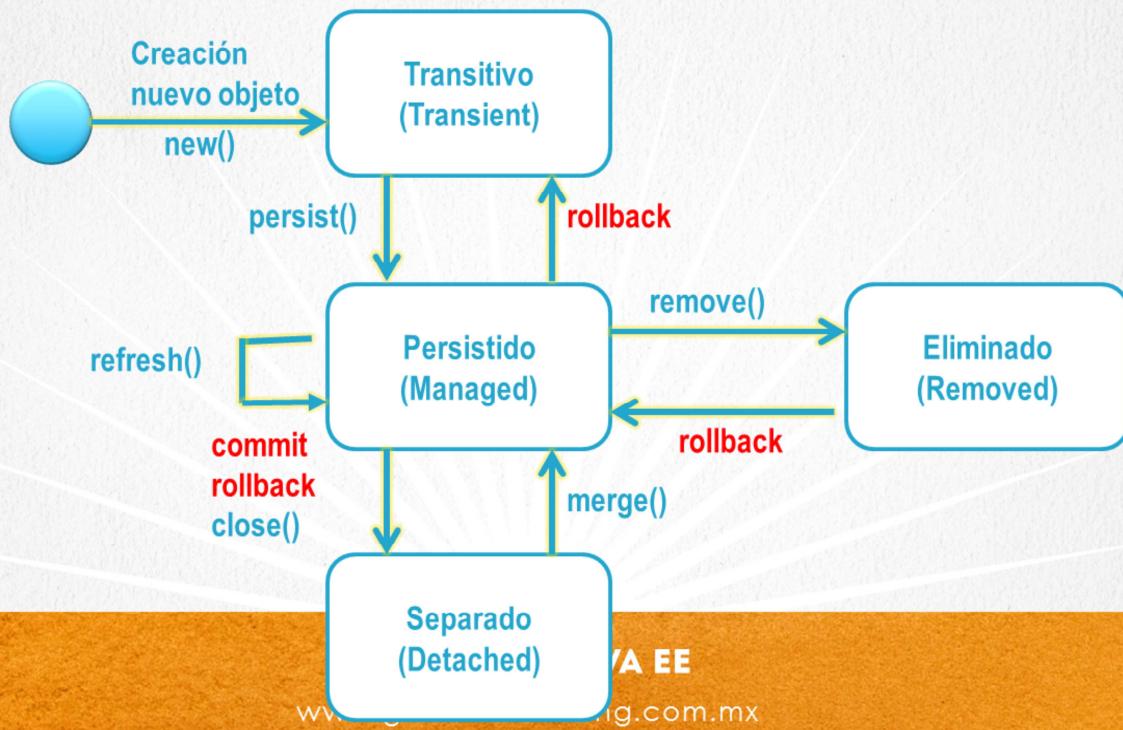
www.globalmentoring.com.mx

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema de Ciclo de Vida en JPA.

¿Estás listo? Ok, ¡Vamos!

CICLO DE VIDA EN JPA



El API de JPA simplifica en gran medida la forma en que interactuamos con una base de datos. JPA agrega un ciclo de vida para la administración de los objetos de entidad. A continuación describimos los estados del ciclo de vida.

Estado Transitivo (Transient):

- ✓ Los objetos de entidad nuevos NO son guardados directamente en la Base de Datos (BD).
- ✓ No están asociados con un registro de BD.
- ✓ Se consideran NO transaccionales.

Estado Persistente (Managed):

- ✓ Un objeto persistente tiene asociado un registro en la BD.
- ✓ Los objetos persistentes siempre están asociados con una transacción. Su estado se sincroniza con la BD al terminar la transacción.

Estado Separado (Detached):

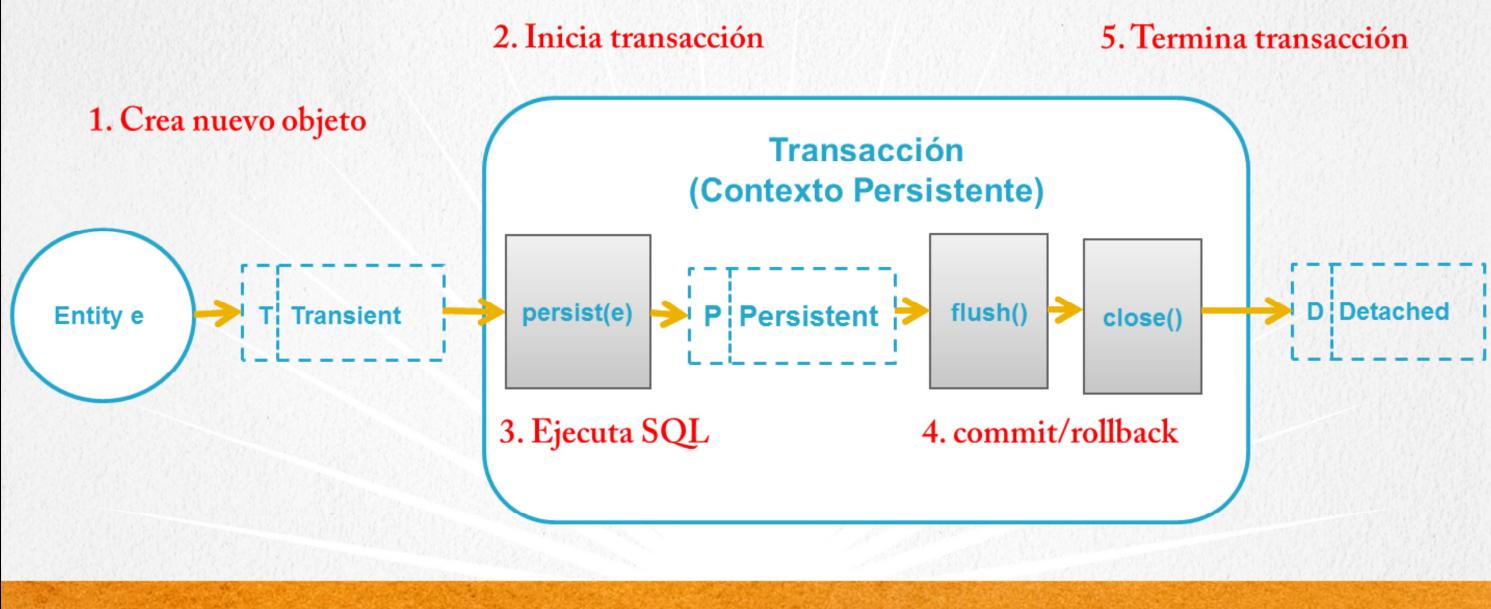
- ✓ Estos objetos tienen asociado un registro de BD, pero su estado no está sincronizado con la BD
- ✓ Todos los objetos recuperados en una transacción se convierten en detached una vez que termina la misma

Estado Eliminado (Removed):

- ✓ Estos objetos ya no tiene una representación con la BD y al terminar la transacción, el registro asociado es eliminado.

Existen además anotaciones para complementar cualquier acción deseada entre los estados mostrados, tales como `@PrePersist` y `@PostPersist`, las cuales se utilizan al persistir un objeto, `@PreRemove` y `@PostRemove` al eliminar un objeto, `@PreUpdate` y `@PostUpdate` al actualizar un objeto, así como `@PostLoad` al hacer refresh del objeto.

PERSISTIR UN OBJETO EN JPA



CURSO DE JAVA EE

www.globalmentoring.com.mx

Código ejemplo para persistir un objeto en JPA:

```

public void testPersistirObjeto() {

    //Paso 1. Crea nuevo objeto
    //Objeto en estado transitivo
    Persona persona1 = new Persona("Pedro", "Luna", null, "pluna@mail.com", "19292943");

    //Paso 2. Inicia transacción
    EntityTransaction tx = em.getTransaction();
    tx.begin();

    //Paso 3. Ejecuta SQL
    em.persist(persona1);

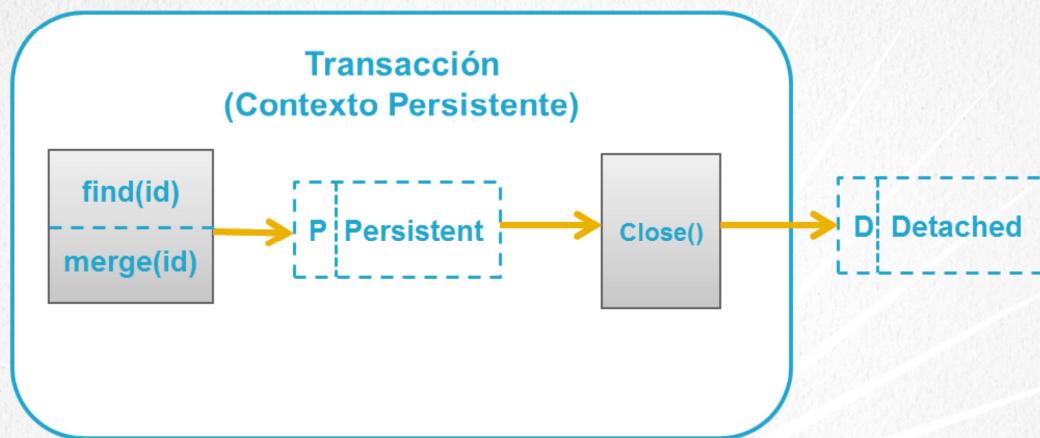
    //Paso 4. Commit/rollback
    tx.commit();

    //Objeto en estado detached
    log.debug("Objeto persistido:" + persona1);
}
  
```

RECUPERAR UN OBJETO DE ENTIDAD EN JPA

1. Inicia transacción

3. Termina transacción



2. Ejecuta SQL

CURSO DE JAVA EE

www.globalmentoring.com.mx

Código de ejemplo para recuperar un objeto en JPA:

```

public void testEncontrarObjeto() {

    //Paso 1. Inicia transacción
    EntityTransaction tx = em.getTransaction();
    tx.begin();

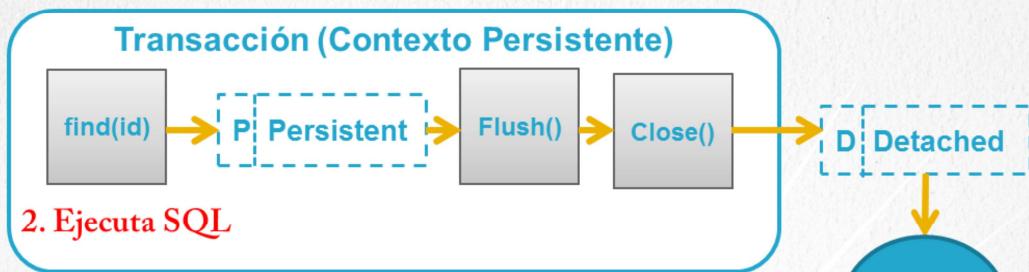
    //Paso 2. Ejecuta SQL de tipo select
    Persona persona1 = em.find(Persona.class, 23);

    //Paso 3. Termina transacción
    tx.commit();

    //Objeto en estado detached
    log.debug("Objeto recuperado:" + persona1);
}
    
```

ACTUALIZAR UN OBJETO PERSISTENTE EN JPA

1. Inicia transacción I



3. Termina transacción I

4. setValue (nuevo valor)

5. Inicia transacción II

Transacción (Contexto Persistente)



7. Termina transacción II

Proceso para actualizar un objeto en JPA:

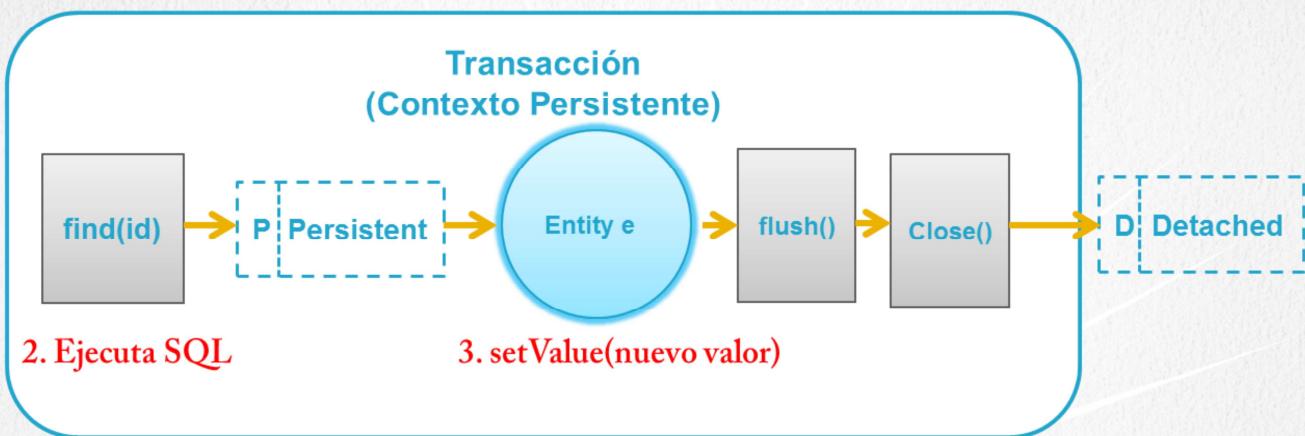
```

public void testActualizarObjeto() {
    //Paso 1. Inicia transacción 1
    EntityTransaction tx1 = em.getTransaction();
    tx1.begin();
    //Paso 2. Ejecuta SQL de tipo select
    //El id proporcionado debe existir en la base de datos
    Persona persona1 = em.find(Persona.class, 23);
    //Paso 3. Termina transacción 1
    tx1.commit();
    //Objeto en estado detached
    log.debug("Objeto recuperado:" + persona1);
    //Paso 4. setValue (nuevoValor)
    persona1.setApeMaterno("Nava");
    //Paso 5. Inicia transacción 2
    EntityTransaction tx2 = em.getTransaction();
    tx2.begin();
    //Paso 6. Ejecuta SQL. Es un select, pero al estar modificado,
    //al terminar la transacción hará un update
    //Como ya tenemos el objeto hacemos solo un merge para resincronizar
    //el objeto a hacer merge, debe contar con el valor de la llave primaria
    em.merge(persona1);
    //Paso 7. Termina transacción 2
    //Al momento de hacer commit, se revisan las diferencias
    //entre el objeto de la base de datos
    //y el objeto en memoria, y se aplican los cambios si los hubiese
    tx2.commit();
    //Objeto en estado detached ya modificado
    log.debug("Objeto recuperado:" + persona1);
}
    
```

ACTUALIZAR UN OBJETO PERSISTENTE CON SESSION LARGA

1. Inicia transacción

4. Termina transacción



Nota: No hay necesidad de hacer un UPDATE explícito, al terminar la transacción (commit) se ejecuta el update.

www.globalmentoring.com.mx

Persistir un objeto en JPA con sesión larga:

```

public void testActualizarObjetoSesionLarga() {

    //Paso 1. Inicia transacción 1
    EntityTransaction tx1 = em.getTransaction();
    tx1.begin();

    //Paso 2. Ejecuta SQL de tipo select
    //Puede ser un find o un merge si ya tenemos el objeto
    Persona persona1 = em.find(Persona.class, 23);

    //Paso 3. setValue (nuevoValor)
    persona1.setApeMaterno("Aguirre");

    persona1.setApeMaterno("Torres");

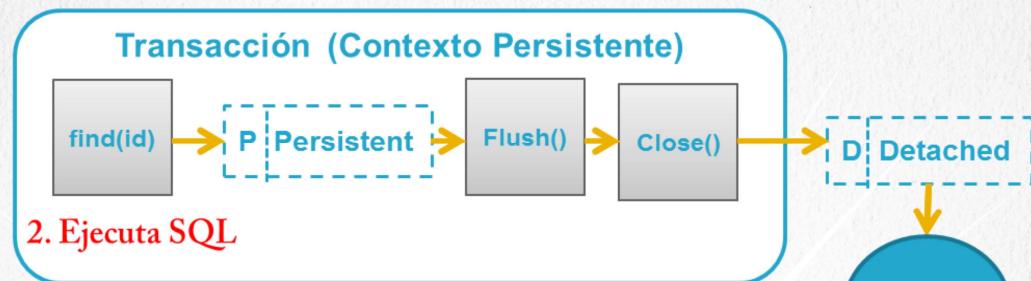
    //Paso 4. Termina transacción 1
    //Ejecuta el update, aunque hayamos hecho 2 cambios sobre el objeto
    //unicamente persiste el último cambio del objeto
    //es decir, el valor de apeMaterno=Torres
    tx1.commit();

    //Objeto en estado detached
    log.debug("Objeto recuperado:" + persona1);
}

```

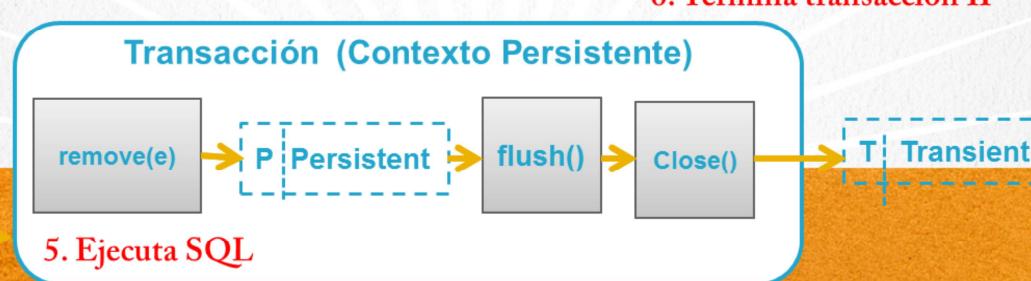
ELIMINAR UN OBJETO PERSISTENTE

1. Inicia transacción I 3. Termina transacción I



4. Inicia transacción II

6. Termina transacción II



Eliminar un objeto en JPA:

```
public void testEliminarObjeto() {
    // Paso 1. Inicia transacción 1
    EntityTransaction tx1 = em.getTransaction();
    tx1.begin();
    // Paso 2. Ejecuta SQL de tipo select
    Persona persona1 = em.find(Persona.class, 23);
    // Paso 3. Termina transacción 1
    tx1.commit();
    // Objeto en estado detached
    log.debug("Objeto recuperado:" + persona1);
    // Paso 4. Inicia transacción 2
    EntityTransaction tx2 = em.getTransaction();
    tx2.begin();
    // Paso 5. Ejecuta SQL (es un delete)
    em.remove(persona1);
    // Paso 6. Termina transacción 2
    // Al momento de hacer commit,
    // se realiza el delete
    tx2.commit();
    // Objeto en estado detached ya modificado
    // Ya no es posible resincronizarlo en otra transacción
    // Solo está en memoria, pero al terminar el método se eliminará
    log.debug("Objeto eliminado:" + persona1);
}
```

EJERCICIOS CURSO HIBERNATE Y JPA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Ciclo de Vida en JPA.

CURSO DE JAVA EE

www.globalmentoring.com.mx

CURSO ONLINE

JAVA EMPRESARIAL JAVA EE

Por: Ing. Ubaldo Acosta

**CURSO DE JAVA EE**www.globalmentoring.com.mx

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- | | |
|---|--|
| <ul style="list-style-type: none">✓ Lógica de Programación✓ Fundamentos de Java✓ Programación con Java✓ Java con JDBC✓ HTML, CSS y JavaScript✓ Servlets y JSP's✓ Struts Framework | <ul style="list-style-type: none">✓ Hibernate Framework & JPA✓ Spring Framework✓ JavaServer Faces✓ Java EE (EJB, JPA y Web Services)✓ JBoss Administration✓ Android con Java✓ HTML5 y CSS3 |
|---|--|

Datos de Contacto:Sitio Web: www.globalmentoring.com.mxEmail: informes@globalmentoring.com.mx