**CONTROL DE CONCURRENCIA EN HIBERNATE**

Para llevar a cabo el control de la concurrencia optimista con hibernate es necesario realizar los siguientes pasos:

1.- Incluir en cada una de las tablas la columna con nombre v (number, valor por defecto 0)

2.- Incluir en cada una de las clases Java que sean objeto de tener control de concurrencia los get y set con la nueva columna llamada v.

**private int v;**

**public int getV() {**

**return v;**

**}**

**public void setV(int v) {**

**this.v = v;**

**}**

3.- A los ficheros hbm correspondientes a cada una de las clases se le añade el tag <versión>

**<version name="v" column="v" unsaved-value="undefined" type="int"/>**

Que automáticamente irá aumentando de valor a medida que se van realizando accesos a una fila concreta de esa tabla.

4.- Una vez que se llama a actualizar a la fila se llama al método updateConcu (en la case HibernateManager) que tiene el siguiente código.

**Método para obtener la versión de la base de datos (número de acceso a esa fila de la tabla).**

**public int getVersionBD (String tabla,String primary, BigDecimal iden) {**

**int num=0;**

**Session session = getSession();**

**try {**

**org.hibernate.Transaction tx = session.beginTransaction();**

**System.out.println("SELECT v FROM "+tabla+" WHERE " + primary + "= " + iden);**

**Query q = session.createSQLQuery("SELECT v FROM "+tabla+" WHERE " + primary + "= " + iden);**

**Object valor = q.uniqueResult();**

**num = Integer.parseInt(valor.toString());**

**} catch (Exception e) {**

**e.printStackTrace();**

**}**

**return num;**

**}**

**Método de actualización concurrente**

Se ha hecho de modo generico pasando un objeto como parámetro para poder reutilizarlo en todos los casos de las clases en la que se aplique la concurrencia. Hago uso del método invoke para invocar el método getV para saber si tenemos una actualización a medias. Si la versión actual es distinta de la que recuperamos de la base de datos lanzamos la excepción StaleObjectStateException que indica que una fila está siendo actualizada por dos instancias al mismo tiempo, mostramos un mensaje y no dejamos que se lleve a cabo la actualización.

**public boolean updateConcu(Object o,String tabla, String primary, BigDecimal ident) throws StaleObjectStateException{**

**// obtener la sesion actual**

**Session session = getSession();**

**Method metodo;**

**try {**

**session.beginTransaction();**

**session.flush();**

**int oldVersion=getVersionBD(tabla,primary,ident);**

**//cargamos el objeto**

**metodo = o.getClass().getMethod("getV");**

**Object result=metodo.invoke(o);**

**int versionactual = Integer.parseInt(result.toString());**

**if (oldVersion != versionactual){**

**FacesMessage msg = new FacesMessage("Otro usuario está actualizando este registro", null);**

**FacesContext.getCurrentInstance().addMessage(null, msg);**

**session.lock(o, LockMode.UPGRADE);**

**throw new StaleObjectStateException(tabla,ident);**

**}**

**session.saveOrUpdate(o);**

**session.getTransaction().commit();**

**} catch (Exception e) {**

**System.out.println("Error en update: " + e);**

**return false;**

**}**

**return true;**

**}**