

# Ejercicios de Persistencia: Introducción a Hibernate

## Índice

1 Ejercicio 1: Instalación y prueba de Hibernate.....	2
2 Ejercicio 2: Hibernate desde Eclipse.....	2

## 1. Ejercicio 1: Instalación y prueba de Hibernate

Vamos a instalar Hibernate en nuestro directorio de trabajo. Para ello se proporcionan tres ficheros que os podéis descargar del área de Recursos de este módulo:

- **hibernate-3.0.5.tar**: contiene las librerías necesarias para poder utilizar Hibernate.
- **hsqldb\_1\_8\_0\_1.zip**: contiene HSQLDB 1.8.0, es una base de datos relacional basada en java y que se almacena en memoria. La utilizaremos para hacer alguna prueba con Hibernate.
- **hibernate-tools-3.1.0.alpha5.zip**: contiene los ficheros para poder utilizar Hibernate desde Eclipse

Para instalar y hacer una pequeña prueba del funcionamiento de Hibernate, seguiremos los siguientes pasos:

1. Descomprimir el fichero **hibernate-3.0.5.tar** en nuestro directorio de trabajo. Esto creará el directorio `hibermante-3.0`.
2. Descomprimir el fichero **hsqldb\_1\_8\_0\_1.zip** en el directorio de trabajo
3. Para comprobar que Hibernate se ha instalado bien, vamos a ejecutar un ejemplo que viene con Hibernate, utilizando para ello la base de datos HSQLDB. Para ello tenemos que copiar el driver JDBC de la base de datos (`hsqldb/lib/hsqldb.jar`) en el directorio `hibernate-3.0/lib`.
4. En el directorio `hibernate-3.0/etc` se encuentra el fichero de propiedades de Hibernate (`hibernate.properties`). Edítalo y comprueba que las propiedades de la base de datos a utilizar son las correspondientes a HSQLDB: simplemente hay que descomentar las propiedades referentes a dicha base de datos (y comentar las del resto). En dicho directorio `etc` también podemos encontrar una plantilla de propiedades Hibernate (`hibernate.properties.template`) con todas las propiedades configurables de Hibernate poder utilizarla en nuestro propio proyecto, comentando y descomentando las que más nos interesen.
5. Desde `hibernate-3.0` teclear `ant eg`. Con esto compilaremos y ejecutaremos el programa ejemplo que se encuentra en el directorio `hibernate-3.0/eg`. Verás que aparecen mensajes de información por pantalla al ejecutar la aplicación. Estos mensajes están precedidos por una indicación sobre el origen del mensaje, por ejemplo: `Environment, Configuration, SettingsFactory,...`
6. Prueba el efecto de la variable `sql_show` poniendo su valor a `true` (fichero `hibernate.properties`).
7. Prueba el efecto de cambiar el valor de la variable `org.hibernate` a `debug` en el fichero `etc/log4j.properties`

## 2. Ejercicio 2: Hibernate desde Eclipse

Vamos a utilizar Hibernate desde Eclipse, para lo cual proporcionamos el proyecto (hibernate-01, en el fichero *hibernate-01.zip*), que tendremos que configurar adecuadamente para poder hacer uso de Hibernate.

Antes de comenzar, vamos a preparar Eclipse para poder utilizar Hibernate, para ello tenemos que Extraer el contenido de **hibernate-tools-3.1.0.alpha5.zip** en el directorio raíz de eclipse mediante la orden `unzip hibernate-tools-3.1.0.alpha5.zip -d $HOME_ECLIPSE`

Una vez que hemos abierto el proyecto en Eclipse, realizaremos lo siguiente:

1. En las propiedades del proyecto, habilitamos hibernate 3.0 mediante:  
*Properties->Hibernate Settings -> Enable Hibernate 3 support.*
2. Añadimos los \*.jar externos necesarios en *Properties->Java build path -> Libraries* desde `hibernate-3.0/lib`. En el directorio `lib` del proyecto tenéis una lista del conjunto mínimo de librerías que se deben incluir para trabajar con Hibernate (realmente, log4j no se requiere pero muchos desarrolladores la prefieren). La librería `hsqldb.jar` está en `hsqldb/lib/`, y la librería `hibernate3.jar` está en el directorio `hibernate-3.0`.
3. Vamos a utilizar la base de datos HSQL, para lo cual crearemos un directorio en el proyecto llamado `data`, en donde la base de datos guardará los ficheros de datos. Acuérdate de enlazar el nuevo *folder* creado con un directorio físico nuevo llamado también `data`.
4. Ahora creamos el fichero de configuración `hibernate.cfg.xml`, mediante *File->new->Other->Hibernate->Hibernate configuration file (cfg.xml)*, con los siguientes valores:
  - Nombre de la Session Factory: `sf01`
  - Dialecto de la base de datos: `HSQL`
  - Clase para el driver: `org.hsqldb.jdbcDriver`
  - Url de la conexión: `jdbc:hsqldb:data/tutorial`
  - Nombre de usuario para la conexión: `sa`.
  - Indicamos, además, que queremos crear una consola de configuración, con los siguientes datos: Nombre de la consola: `hibernate-01`. En el *classpath*, añadir el fichero `jar` de la BD. En las propiedades del proyecto, añadiremos la consola de configuración mediante *Properties->Hibernate Settings -> Default Hibernate Console Configuration -> hibernate-01*.
  - Añadimos las propiedades `show_sql` con valor `true`, y `hbm2ddl.auto` con el valor `create` (para que cree la base de datos de nuevo con cada ejecución del

programa.

- También añadimos el recurso de correspondencia (*mapping resource* `Event.hbm.xml`). Con esto ya tenemos creado el fichero de configuración de Hibernate, que deberá estar situado en el directorio `src`.

Ya tenemos casi todo listo para ejecutar la aplicación. Nos queda (aunque esto es opcional) copiar el fichero `hibernate-3.0/etc/log4j.properties` en `src`. Esto configura los mensajes de salida de Hibernate. Podemos editar dicho fichero para ver que hemos activado los mensajes de tipo `info`.

Para probar la aplicación cargamos el fichero `build.xml` y ejecutamos los *targets compile* y *run* en este orden.

Finalmente, realiza los siguientes cambios en el programa:

1. Prueba a cambiar la base de datos para utilizar MySQL en lugar de HQL. Para ello tienes que cambiar la configuración de la base de datos. Puedes consultar la plantilla de propiedades de Hibernate (`hibernate-3.0/etc/hibernate.properties.template`). Para utilizar MySQL en el laboratorio, podéis utilizar un usuario y base de datos ya creados que son usuario: `tw`, password: `tw`, base de datos: `foros`.
2. ¿Qué le ocurre a la BD entre dos ejecuciones consecutivas? ¿Por qué ocurre esto? Realiza los cambios necesarios para poder añadir nuevos eventos en varias ejecuciones consecutivas.

