

Hibernate_II

Primeros programas

Librerías necesarias

- Con la configuración realizada hasta ahora no se puede programar en Java, es necesario añadir una serie de paquetes al proyecto. Para ello:
 - Accedemos a <http://hibernate.org/orm/downloads/> y nos descargamos la última distribución estable de Hibernate. Descomprimos el ZIP. Nos quedará una carpeta con un nombre parecido a **hibernate-release-<version>**
 - Creamos una carpeta -en el Escritorio, por ejemplo- que llamaremos Hibernate. Nos colocamos en la carpeta **hibernate-release-<version>/lib/require** y copiamos todos los archivos a la carpeta recién creada en el Escritorio.
 - Descargamos la última versión de **slf4j** desde <https://www.slf4j.org/download.html>.
 - Descomprimos y localizamos los ficheros **slf4j-api-<version>.jar** y **slf4j-simple-<version>.jar**. Los copiamos en la carpeta que hemos llamado Hibernate

- Desde Eclipse hacemos clic con el botón derecho del ratón en nuestro proyecto y pulsamos **Build Path** → **Add Libraries** → **User Library**. Pulsamos “User Libraries” y después “New”
 - Nos pedirá un nombre para la librería que queremos agregar. Escribimos por ejemplo: **Hibernate.lib**
 - Pulsamos el botón “Add External JARS”. Seleccionamos todos los ficheros de la carpeta Hibernate.
- Ahora ya estamos en condiciones de poder trabajar.
 - En primer lugar necesitaremos crear un **Singleton**.
 - El **Singleton** es un patrón diseñado para restringir la creación de objetos pertenecientes a una clase. Su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella.
 - Nuestro Singleton será una clase de ayuda que accede a **SessionFactory** para obtener un objeto de sesión, hay una única **SessionFactory** para toda la aplicación. El nombre de la clase es *HibernateUtil.java* y se incluirá en el paquete *primero* de nuestro proyecto. El código es:

```
package primero;
import org.hibernate.SessionFactory;
import org.hibernate.boot.registry.StandardServiceRegistryBuilder;
import org.hibernate.cfg.Configuration;
public class HibernateUtil {
//Para asegurar que la clase no pueda ser instanciada nuevamente se regula el alcance del constructor con el atributo
//private
    private static final SessionFactory sessionFactory = buildSessionFactory();
    private static SessionFactory buildSessionFactory(){
        try{
            return new Configuration().configure().buildSessionFactory(new
StandardServiceRegistryBuilder().configure().build());
        }
        catch (Throwable ex) {
            System.err.println("Initial SessionFactory creation failed."+ex);
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory(){
        return sessionFactory;
    }
}
```

- Con la clase anterior podemos obtener la sesión actual desde cualquier parte de nuestro proyecto.
- Ahora añadimos a nuestro proyecto una nueva clase que llamaremos *Main.java*. Esta se creará en el *default package* del proyecto. Esta clase insertará una fila en la tabla *depart*

```
import primero.*;
import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;
public class Main {
    public static void main(String[] args){
//En primer lugar se obtiene la sesión creada por el Singleton.
        SessionFactory session = HibernateUtil.getSessionFactory();
//Abrimos sesión e iniciamos una transacción
        Session session = session.openSession();
        Transaction tx = session.beginTransaction();
        System.out.println("Inserto una fila en depart");
//Creamos un nuevo objeto Depart y damos valor a sus atributos
        Depart dep = new Depart();
        dep.setDeptNo((byte) 62);
        dep.setDnombre("MARKETING");
        dep.setLoc("GUADALAJARA");
//Guardamos en la base de datos y comprometemos la información
        session.save(dep);
        tx.commit();
        session.close();
        System.exit(0);
    }
}
```

Problemas Hibernate_II

- Realiza el ejemplo expuesto anteriormente. Comprueba que añade una nueva fila en depart
- Realiza un programa Java (puedes modificar el Main anterior) que añada una nueva fila en la tabla emple. El nuevo empleado estará asignado al departamento 10. Para asignarle el departamento será necesario crear un objeto de tipo Depart y asignar como número de departamento el valor 10 con el método *setDeptNo()*.