Práctica No. 1. Instalación y Conexión a Bases de Datos con Hibernate

Preparación de la Base de Datos

- Descargar de la página web <u>http://academicos.azc.uam.mx/jfg/pags/tarea_taller_web.html</u> el archivo *script_01_generacion_clase.sql* el cuál contiene la base de datos y las tablas a utilizar.
- Abrir el entorno MySQL Workbench y ejecutar el script para crear la base de datos (no es necesario crear las tablas).

Bibliotecas Necesarias

- Ingresar a la página <u>http://hibernate.org/</u> para descargar las bibliotecas necesarias de Hibernate.
- Dar clic en *More* de *Hibernate ORM*



• Dar clic en Download (5.2.10.Final) para descargar la última versión



- Se descargará un archivo con bibliotecas y documentación, al descomprimirlo se pueden presentar errores por la longitud de los nombres de archivo de la documentación. En particular interesan los archivos dentro del directorio *lib/required*
- Ingresar a la página <u>https://www.mysql.com/products/connector/</u> para descargar el *driver* para la conexión con MySQL y seleccionar la liga *JDBC Driver for MySQL*



- Descargar la versión ZIP o TAR según se desee
- Descomprimir el archivo, solo interesa el archivo *mysql-connector-java-5.1.44-bin.jar*

Entorno de Desarrollo

- Abrir el entorno de desarrollo Eclipse
- Se creará un nuevo proyecto Java llamado *InstalacionEclipse* (asegurarse que se tiene la versión 1.8 de JDK)
- Se creará un nuevo directorio llamado *lib* (*Clic derecho sobre el proyecto* → *New* → *Folder*)
- Copiar los archivos del directorio *lib/required* al directorio *lib* en el proyecto
- Copar el archivo del driver de MySQL, *mysql-connector-java-5.1.44-bin.jar*, al directorio *lib* del proyecto
- Crear los siguientes paquetes:
 - hibernate.principal
 - hibernate.conexion
- Para que se reconozcan las bibliotecas necesarias, se deben agregar al proyecto
- Clic derecho sobre el proyecto \rightarrow *Properties*



- Seleccionar Add External JARs
- Buscar el directorio lib en donde se copiaron las bibliotecas, seleccionarlas todas y dar clic en *Aceptar*

type filter text 🛛 🕱	Java Build Path	(¬ = ¬ = −
Resource Builders	Source Projects Libraries Order and Export	
Java Build Path ▶ Java Code Style ▶ Java Compiler ▶ Java Editor	JARs and class folders on the build path:	Add JARs
	 classmate-1.3.0.jar - /home/josue/workspace_EE/Ins dom4j-1.6.1.jar - /home/josue/workspace_EE/Instala 	Add External JARs
Javadoc Location	ibernate-commons-annotations-5.0.1.Final.jar - /ho	Add Variable
Project Facets Project References	 Hibernate-core-5.2.10.Final.jar - /home/josue/works; Hibernate-ipa-2.1-api-1.0.0.Final.jar - /home/josue/works; 	Add Library
Run/Debug Settings	jandex-2.0.3.Final.jar - /home/josue/workspace_EE/li	Add Class Folder
Task Repository Task Tags	 Javassist-3.20.0-GA.jar - /home/josue/workspace_EE/ Jboss-logging-3.3.0.Final.jar - /home/josue/workspac 	Add External Class Folder.
 Validation WikiText 	jboss-transaction-api_1.2_spec-1.0.1.Final.jar - /home mvsql-connector-java-5.1.44-bin.jar - /home/josue/w	Edit
	IRE System Library [JavaSE-1.8]	Remove
		Migrate JAR File

• Finalmente, dar clic en OK para que queden cargadas en el proyecto

Configuración y Prueba de la Conexión

Se debe crear un archivo con los datos para la conexión con la Base de Datos usando Hibernate

Clic Derecho sobre $src \rightarrow New \rightarrow Other$ Seleccionar de las opciones XML $\rightarrow XML$ *File* $\rightarrow Next$ En el nombre escribir *hibernate.cfg.xml*

```
hibernate.cfg.xml
```

```
<?xml version='1.0' encoding='utf-8'?>
<hibernate-configuration>
  <session-factory>
     <property name="hibernate.connection.driver class"></pro>
      com.mysql.jdbc.Driver
       </property>
     <property name="hibernate.connection.url"></property name="hibernate.connection.url">
       jdbc:mysql://localhost:3306/practicas taller web
       </property>
     <property name="hibernate.connection.username"></property name="hibernate.connection.username">
       root
       </property>
     <property name="hibernate.connection.password"></pro>
      root
       </property>
     <property name="hibernate.dialect"></property name="hibernate.dialect">
      org.hibernate.dialect.MySQLDialect
       </property>
  </session-factory>
</hibernate-configuration>
```

Conexión

• Crear la clase *CrearConexion* en el paquete *hibernate.conexion*

CrearConexion.java

```
package hibernate.conexion;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;
public class CrearConexion {
    private static final SessionFactory sessionFactory =
    buildSessionFactory();
    private static SessionFactory buildSessionFactory() {
```

```
try {
    // Crea una sesion basado en hibernate.cfg.xml
    return new
Configuration().configure().buildSessionFactory();
    }
    catch (Throwable ex) {
      System.err.println("Error al crear sessionFactory " + ex);
      throw new ExceptionInInitializerError(ex);
      }
    public static SessionFactory getSessionFactory() {
      return sessionFactory;
    }
}
```

Crear la clase Principal en el paquete hibernate.principal

```
Principal.java
package hibernate.principal;
import hibernate.conexion.CrearConexion;
import org.hibernate.Session;
public class Principal {
     public static void main(String[] args) {
           Session sesion =
CrearConexion.getSessionFactory().openSession();
        String sql = "SELECT VERSION()";
           String resultado = (String)
sesion.createNativeQuery(sql).getSingleResult();
           System.out.println("La versión que estás usando es: " +
resultado);
        CrearConexion.getSessionFactory().close();
      }
}
```

Aparecerá el mensaje, lo que indica que la conexión ha sido exitosa.