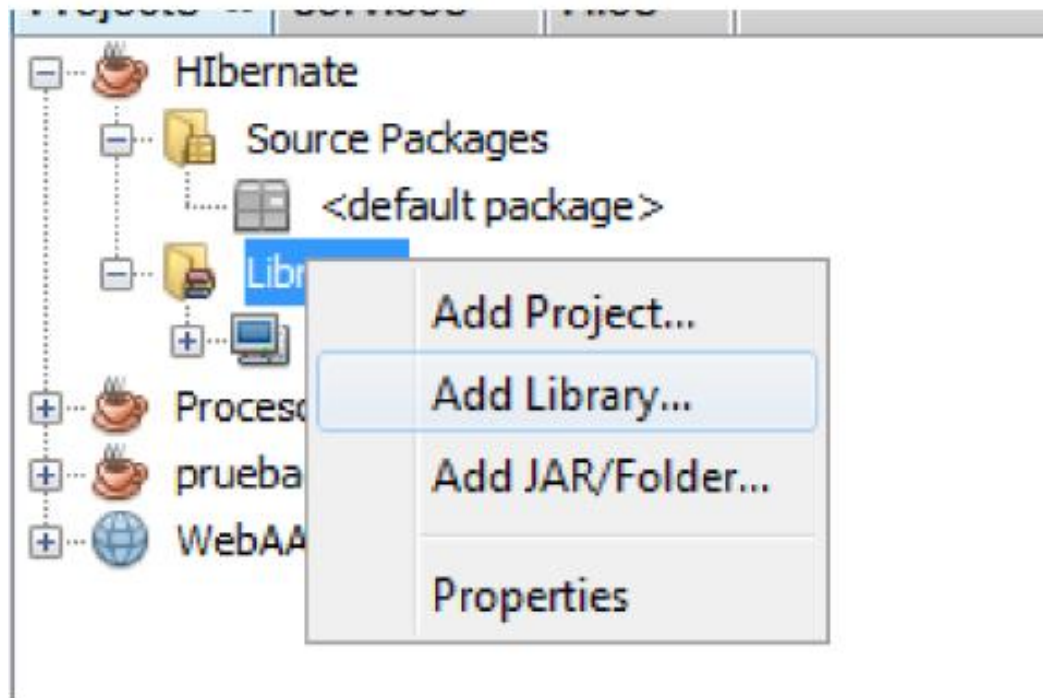


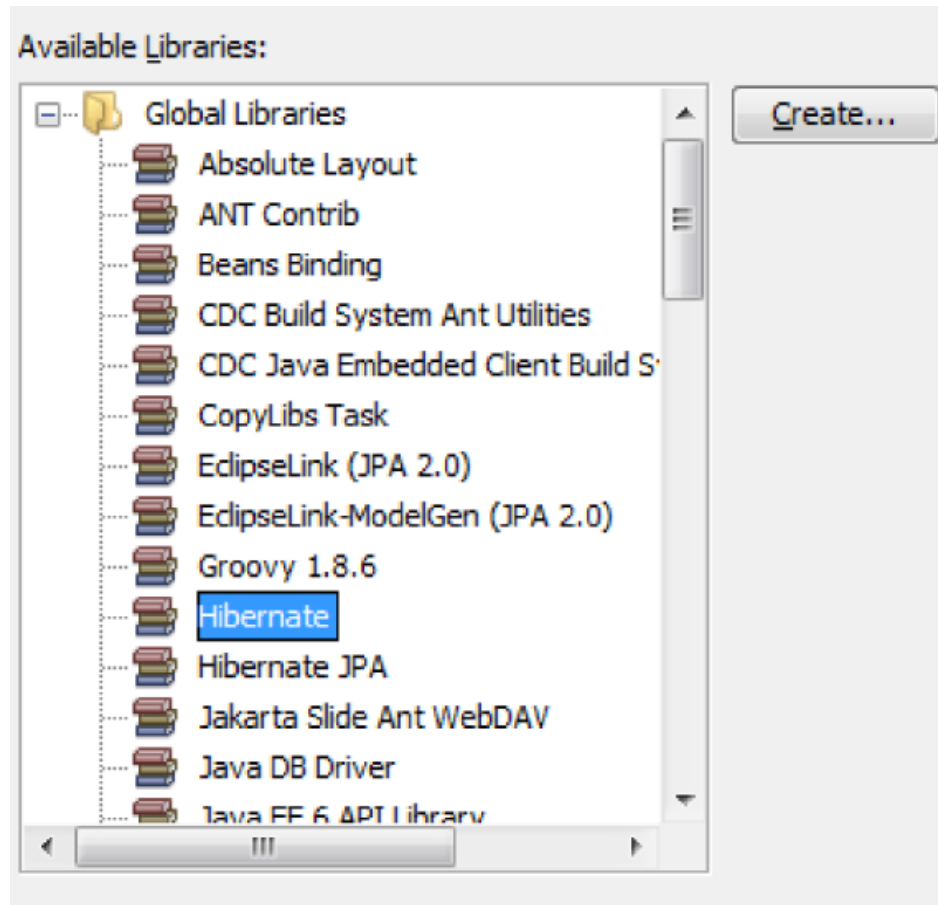
HIBERNATE

Acceso a datos

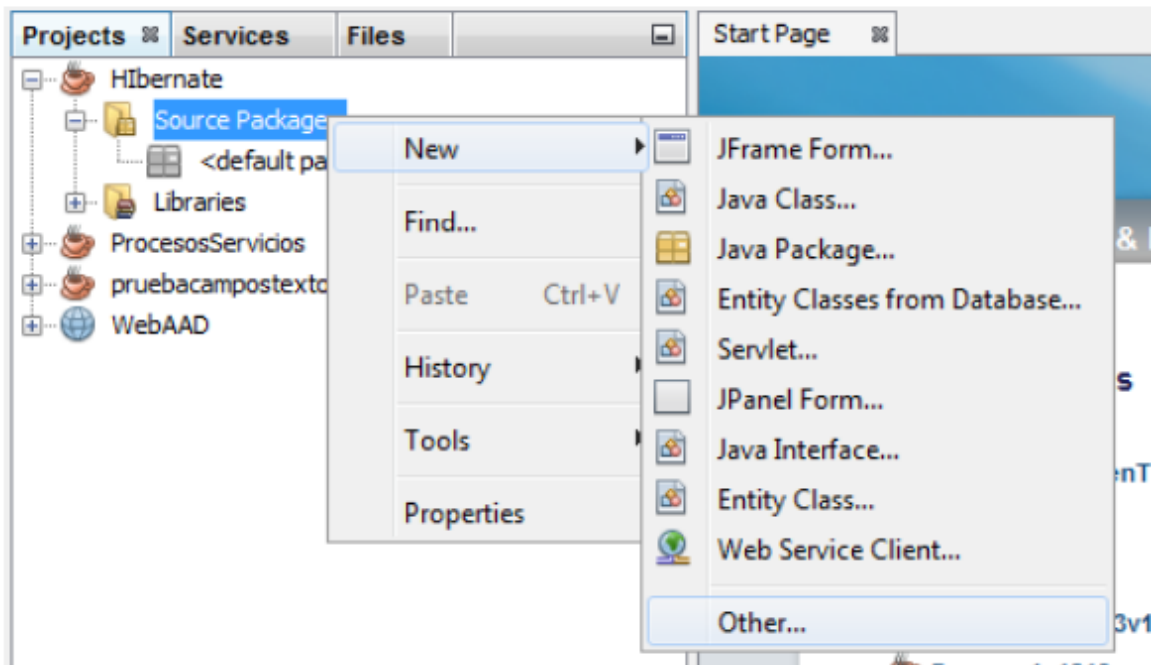
AÑADIR HIBERNATE A UN PROYECTO



SELECCIONAR LA LIBRERÍA



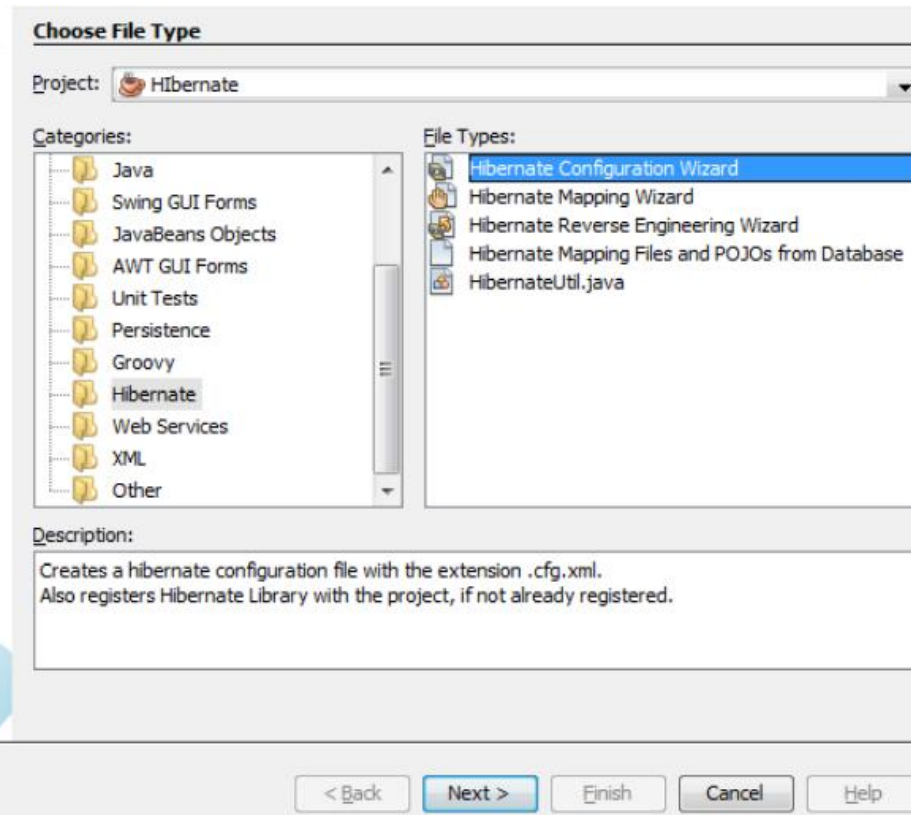
CONFIGURACIÓN



ARCHIVO DE CONFIGURACIÓN DE HIBERNATE

Steps

1. Choose File Type
2. ...



HIBERNATE.CFG

Name and Location

File Name:

Project:

Folder:

Created File:



CONFIGURAR CONEXIÓN

Customize Connection

Driver Name: MySQL (Connector/J driver) ▼

Host: localhost Port: 3306

Database: hibernate


User Name: hibernate

Password: ●●●●●●●●

☒ Remember password

Test Connection

JDBC URL: jdbc:mysql://localhost:3306/hibernate?zeroDateTimeBehavior=convertToNull

 Connection Succeeded.



SELECCIONAR LA CONEXIÓN

Select Data Source






Database Connection: jdbc:mysql://localhost:3306/hibernate?zeroDateTimeBehavior=convertTo...

Database Dialect: org.hibernate.dialect.MySQLDialect



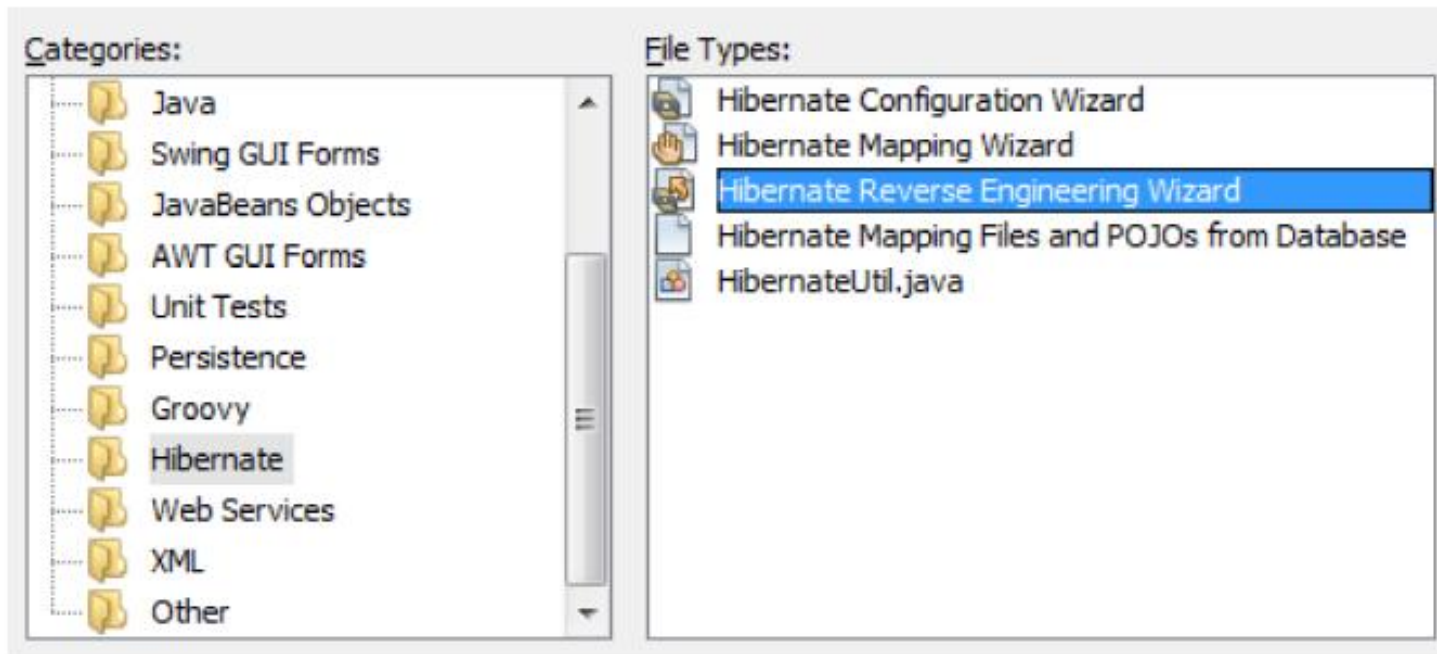
HIBERNATEUTIL.JAVA

File Types:

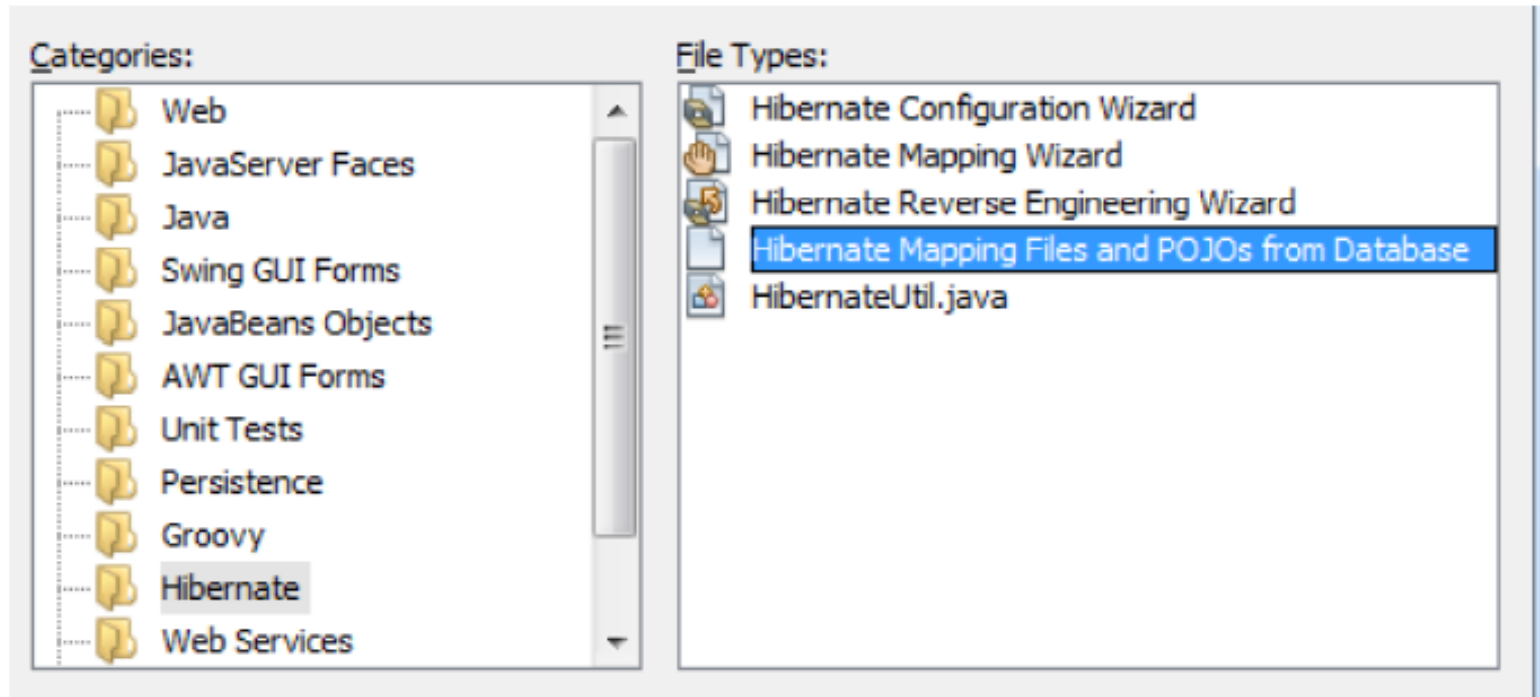
-  Hibernate Configuration Wizard
-  Hibernate Mapping Wizard
-  Hibernate Reverse Engineering Wizard
-  Hibernate Mapping Files and POJOs from Database
-  **HibernateUtil.java**



ARCHIVO DE INGENIERÍA INVERSA



ARCHIVOS POJO Y DE MAPEO



GENERACIÓN DE CÓDIGO

New Hibernate Mapping Files and POJOs from Database

Steps

1. Choose File Type
2. **Generation of Code**

Generation of Code

Hibernate Configuration File:

Hibernate Reverse Engineering File:

General Settings:

- ☐ JDK 5 Language Features
- ☐ EJB 3 Annotations

Code Generation Settings:

- ☒ Domain Code (.java)
- ☒ Hibernate XML Mappings (.hbm.xml)

Project:

Location:

Package:

< Back Next > **Finish** Cancel Help

CLASES

- Se utiliza un objeto **SessionFactory** para obtener instancias de Session.
- Las instancias de **Session** se utilizan para operar contra la base de datos.
- Se utiliza una instancia de **Transaction** para inicializar y terminar las operaciones contra la base de datos.
- Se utiliza un objeto de la clase **Query** para lanzar las consultas contra la base de datos.
- El lenguaje de consulta que se usa para las consultas es **HQL**.



ACTUALIZACIONES

```
Mesa m = (Mesa) session.load(Mesa.class,  
                               Integer.parseInt(id));
```

```
session.delete(m);
```

```
session.update(m);
```

```
session.save(m);
```



TRANSACCIONES

```
session.beginTransaction();
```

```
session.getTransaction().commit();
```

```
session.getTransaction().rollback();
```



BORRADO

```
session = HibernateUtil.getSessionFactory().  
                                openSession();  
session.beginTransaction();  
Mesa m = (Mesa) session.load(Mesa.class,  
                                Integer.parseInt(id));  
session.delete(m);  
session.getTransaction().commit();  
session.flush();  
session.close();
```



EDICIÓN

```
session.beginTransaction();  
Mesa m = (Mesa) session.load(Mesa.class,  
                               Integer.parseInt(id));  
m.setNombre(mesa);  
session.update(m);  
session.getTransaction().commit();
```



INSERCIÓN

```
session.beginTransaction();  
Mesa m = new Mesa(mesa);  
session.save(m);  
session.getTransaction().commit();
```



SELECCIÓN I

```
session.beginTransaction();  
String hql = "from Mesa";  
Query q = session.createQuery(hql);  
List<Mesa> mesas = q.list();  
session.getTransaction().commit();
```



SELECCIÓN II

```
session.beginTransaction();  
String hql = "from Mesa";  
Query q = session.createQuery(hql);  
q.setFetchSize(10);  
Iterator i = q.iterate();  
while (i.hasNext()){  
    m = (Mesa) i.next();  
}  
session.getTransaction().commit();
```



CONSULTAS PREPARADAS

- parámetros anónimos ?

```
String hql = "from Mesa where id > ?";  
Query q = session.createQuery(hql);  
q.setInteger(0, Integer.parseInt(id));
```

- parámetros con nombre **:nombre**

```
String hql = "from Mesa where id > :id";  
Query q = session.createQuery(hql);  
q.setInteger("id", Integer.parseInt(id));
```



LOAD() Y GET()

- `load()`, se usa cuando se tiene certeza de que existe el registro que cumple la condición, si no existe se lanza una excepción;
- `Mesa m = (Mesa) session.load(Mesa.class, Integer.parseInt(id));`
- `get()`, se usa cuando no se tiene certeza de que existe el registro, si no existe devuelve `null`;
- `Mesa m = (Mesa) session.get(Mesa.class, Integer.parseInt(id));`



CLAVES FORÁNEAS

```
Pedido p = (Pedido) session.load(Pedido.class,  
                                   Integer.parseInt(id));  
Set<Detallepedido> lista = p.getDetallepedidos();  
Iterator<Detallepedido> i = lista.iterator();  
Detallepedido dp;  
while(i.hasNext()){  
    dp = i.next();  
    ...  
}
```



CONSULTAS DE UN SOLO REGISTRO

```
String hql = "from Mesa where id = ?";  
Query q = session.createQuery(hql);  
q.setInteger(0, Integer.parseInt(id));  
m = (Mesa) q.uniqueResult();
```



CONSULTAS SOBRE MÁS DE UNA TABLA

```
String hql = "from Mesa m join Pedido p on  
m.id=p.mesa ";
```

```
Query q = session.createQuery(hql);
```

```
Iterator i = q.iterate();
```

```
while(i.hasNext()){
```

```
    Object[] o = (Object[]) i.next();
```

```
    m = (Mesa) o[0];
```

```
    p = (Pedido) o[1];
```

```
}
```



CONSULTAS DE VALORES

```
hql = "select m.id, p.cerrado from Mesa m, Pedido p  
where m.id = p.mesa";//sql tradicional
```

```
hql = "select m.id, p.cerrado from Pedido p inner join  
p.mesa m ";//sql moderno
```

```
q = session.createQuery(hql);
```

```
i = q.iterate();
```

```
while(i.hasNext()){
```

```
    Object[] o = (Object[]) i.next();
```

```
    int v1 = (Integer)o[0];
```

```
    byte v2 = (Byte)o[1];
```

```
}
```



UPDATE Y DELETE CON HQL

```
String hql = "update Mesa m set m.nombre = 'valor'
             where m.id>10 ";
```

```
Query q = session.createQuery(hql);
int r = q.executeUpdate();
```

```
String hql = "delete Mesa m where m.id>10 ";
Query q = session.createQuery(hql);
int r = q.executeUpdate();
```



INSERT EN HQL

```
String hql = "insert into Mesa (nombre) select  
    concat(id, nombre) from Mesa ";  
Query q = session.createQuery(hql);  
int r = q.executeUpdate();
```

