

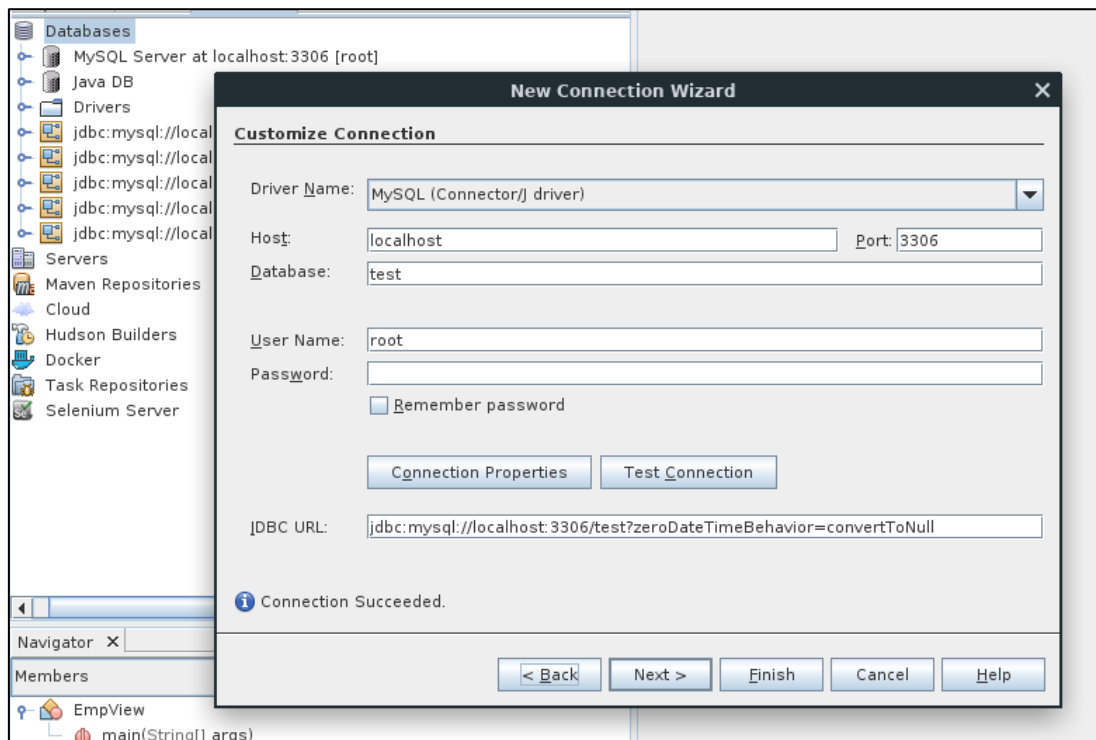
Practica Hibernate UserCRUD

Por Jorge María Huguet

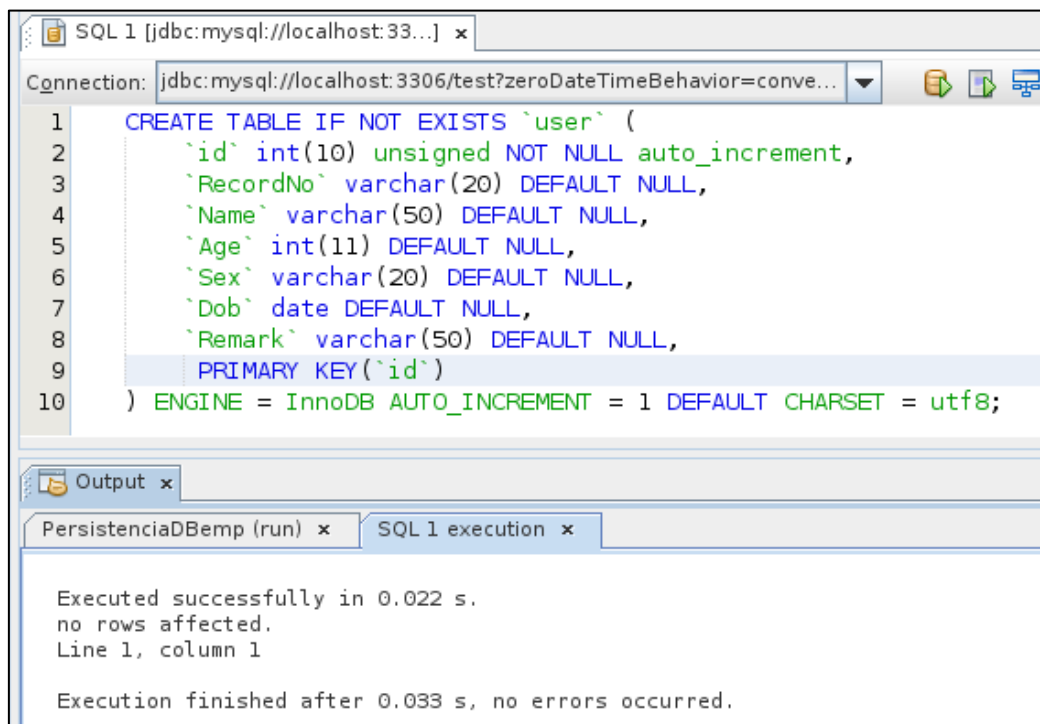
Índice

1.	Se crea la conexión.....	3
2.	Se crea la tabla y se incluyen registros.....	3
3.	Se incluyen los registros	4
4.	Se crea el proyecto	4
5.	Añadimos el driver del conector Mysql.....	6
6.	Se crean los packages y sus clases.....	6
6.1.	Creating Reverse Engineering File-> hibernate.reveng.xml	7
6.2.	Creating Hibernate Entity (pojo) File: user.java.....	8
6.3.	Creating JSF Managed Bean File UserBean.java.....	8
6.4.	Creating DataAccessObject (DAO) File UserDAO.java	9
6.5.	HibernateUtil.java	9
6.6.	Creating index.xhtml.....	10
6.7.	Search.xhtml	11
6.8.	UserCRUD.xhtml	11
6.9.	web.xml (Automatically generated).....	12
7.	Se ejecuta el programa.....	12
7.1.	Insertar User	13
7.2.	Buscar User.....	13
7.3.	Modificar User	14
7.4.	Borrar User	14

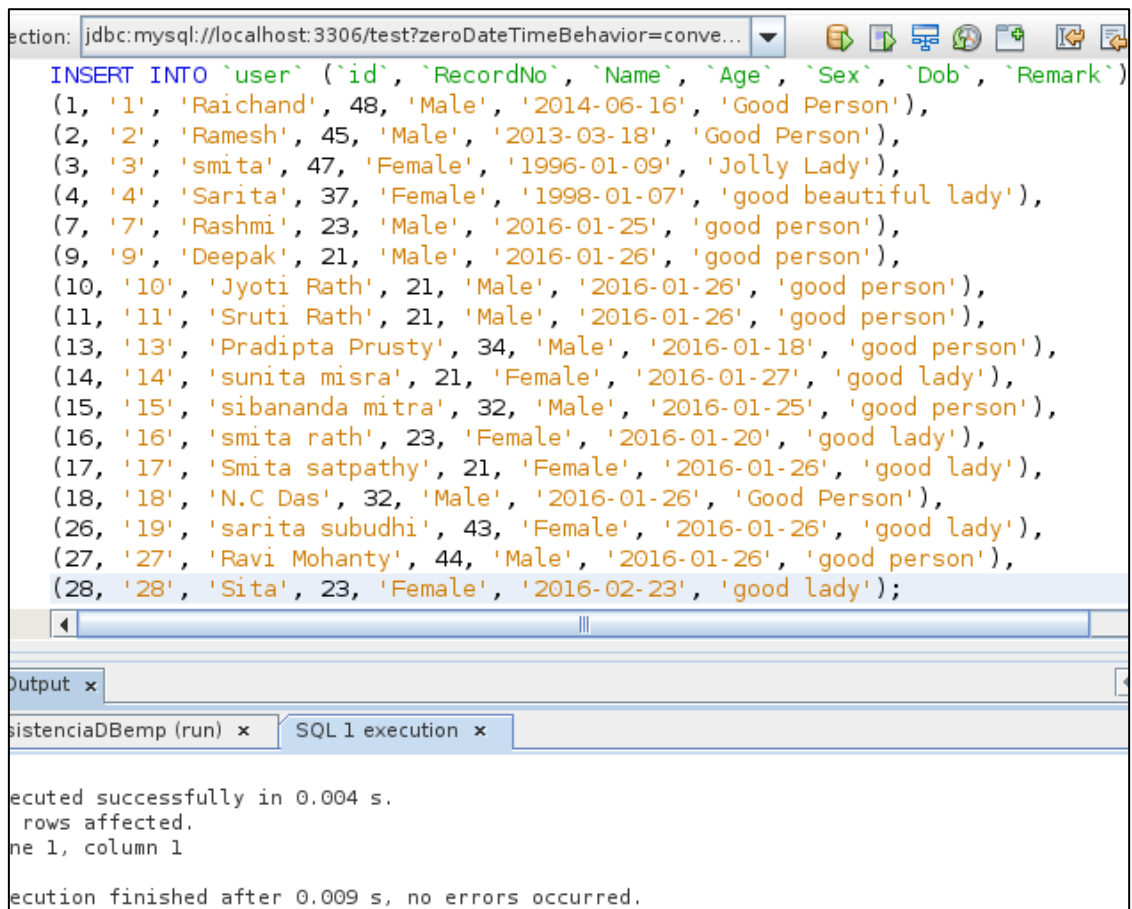
1. Se crea la conexión



2. Se crea la tabla y se incluyen registros



3. Se incluyen los registros



The screenshot shows a SQL IDE window with a connection string: `jdbc:mysql://localhost:3306/test?zeroDateTimeBehavior=conve...`. The main editor contains an `INSERT INTO` statement for a table named `user`. The statement lists 28 records with columns: `id`, `RecordNo`, `Name`, `Age`, `Sex`, `Dob`, and `Remark`. The records include names like Raichand, Ramesh, smita, Sarita, Rashmi, Deepak, Jyoti Rath, Sruti Rath, Pradipta Prusty, sunita misra, sibananda mitra, smita rath, Smita satpathy, N.C Das, sarita subudhi, Ravi Mohanty, and Sita. The output pane at the bottom shows the execution was successful in 0.004 seconds, affecting 28 rows, and finished after 0.009 seconds with no errors.

```
INSERT INTO `user` (`id`, `RecordNo`, `Name`, `Age`, `Sex`, `Dob`, `Remark`)
(1, '1', 'Raichand', 48, 'Male', '2014-06-16', 'Good Person'),
(2, '2', 'Ramesh', 45, 'Male', '2013-03-18', 'Good Person'),
(3, '3', 'smita', 47, 'Female', '1996-01-09', 'Jolly Lady'),
(4, '4', 'Sarita', 37, 'Female', '1998-01-07', 'good beautiful lady'),
(7, '7', 'Rashmi', 23, 'Male', '2016-01-25', 'good person'),
(9, '9', 'Deepak', 21, 'Male', '2016-01-26', 'good person'),
(10, '10', 'Jyoti Rath', 21, 'Male', '2016-01-26', 'good person'),
(11, '11', 'Sruti Rath', 21, 'Male', '2016-01-26', 'good person'),
(13, '13', 'Pradipta Prusty', 34, 'Male', '2016-01-18', 'good person'),
(14, '14', 'sunita misra', 21, 'Female', '2016-01-27', 'good lady'),
(15, '15', 'sibananda mitra', 32, 'Male', '2016-01-25', 'good person'),
(16, '16', 'smita rath', 23, 'Female', '2016-01-20', 'good lady'),
(17, '17', 'Smita satpathy', 21, 'Female', '2016-01-26', 'good lady'),
(18, '18', 'N.C Das', 32, 'Male', '2016-01-26', 'Good Person'),
(26, '19', 'sarita subudhi', 43, 'Female', '2016-01-26', 'good lady'),
(27, '27', 'Ravi Mohanty', 44, 'Male', '2016-01-26', 'good person'),
(28, '28', 'Sita', 23, 'Female', '2016-02-23', 'good lady');
```

Output x

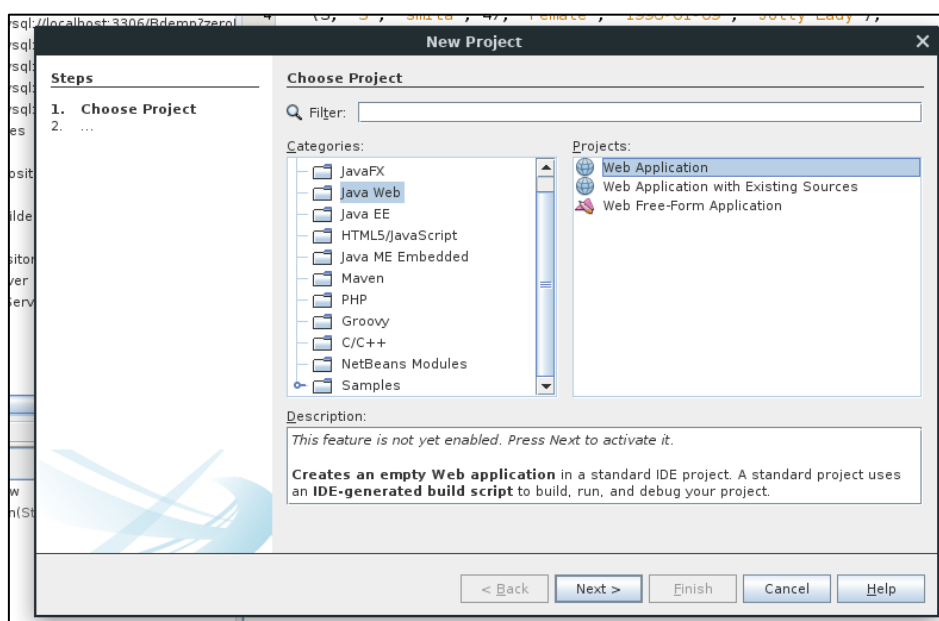
SQL 1 execution x

ecuted successfully in 0.004 s.
rows affected.
ne 1, column 1

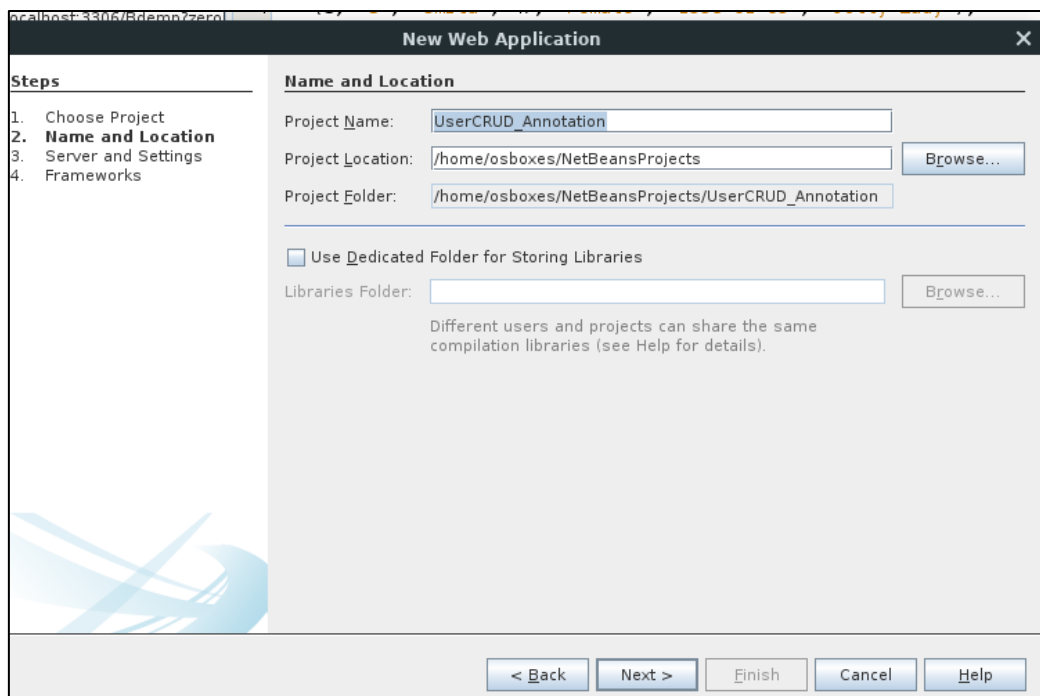
ecution finished after 0.009 s, no errors occurred.

4. Se crea el proyecto

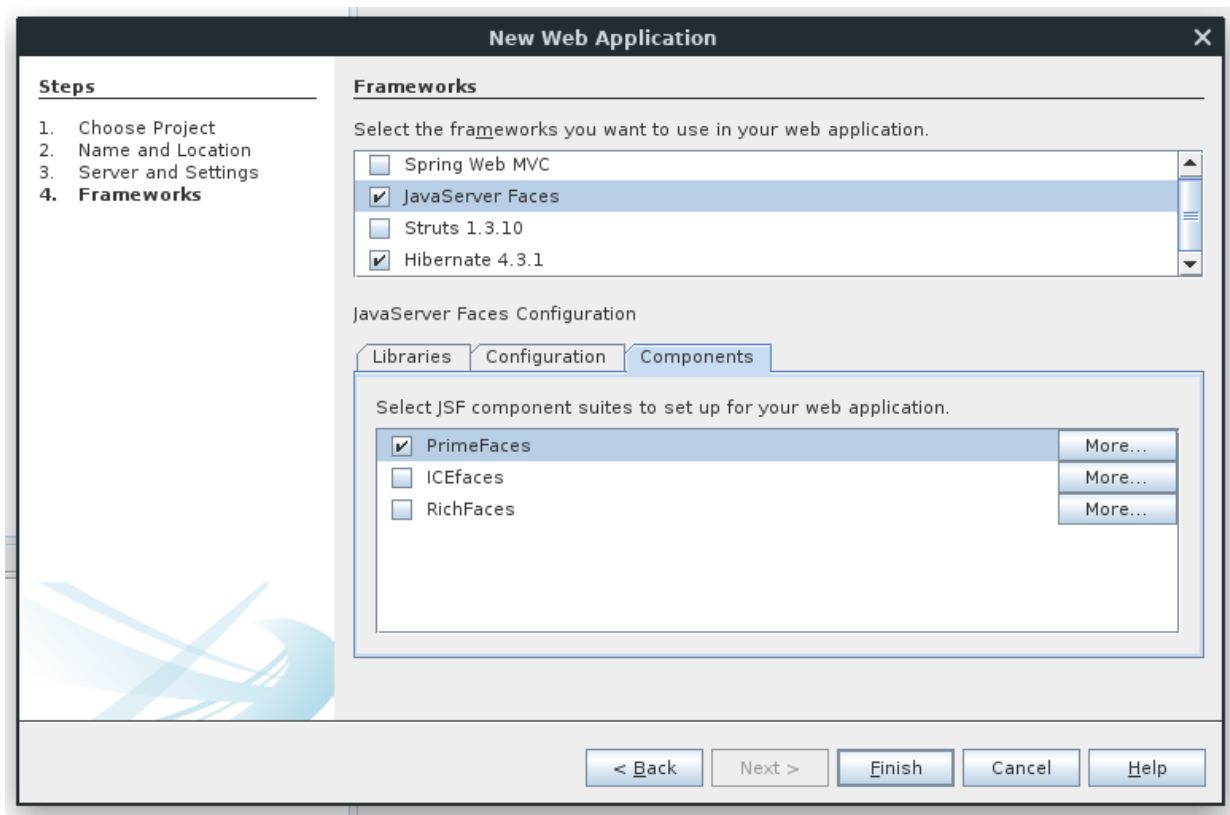
En categoría seleccionamos Java Web y en proyecto Web Application.



Damos un nombre al proyecto

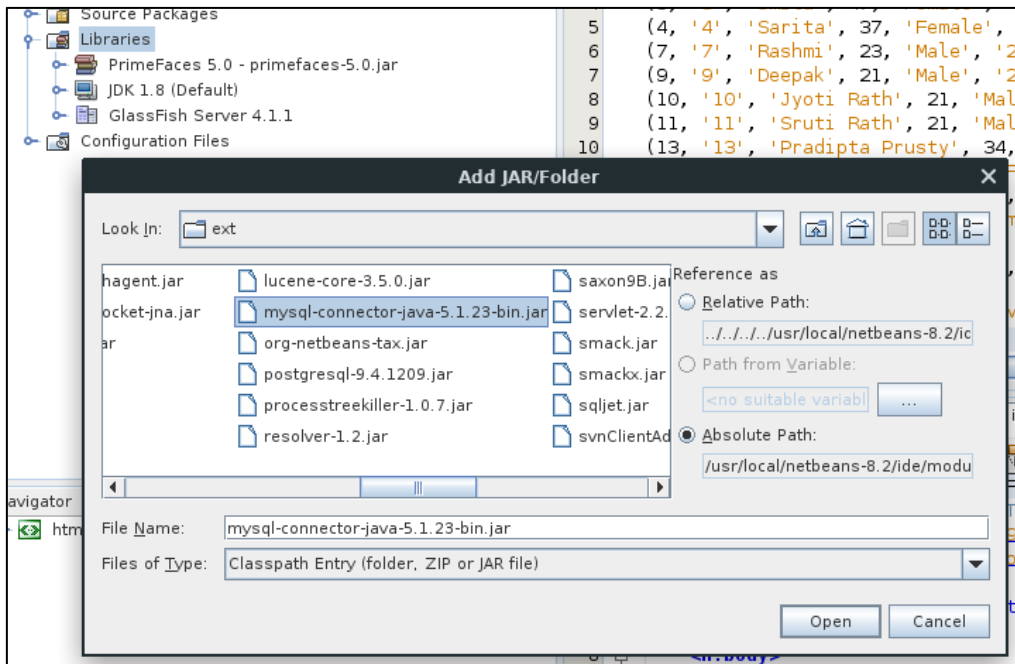


En Frameworks seleccionamos Hibernate 4.3.1 y después JavaServer Faces, y en la pestaña Components pulsamos PrimeFaces.



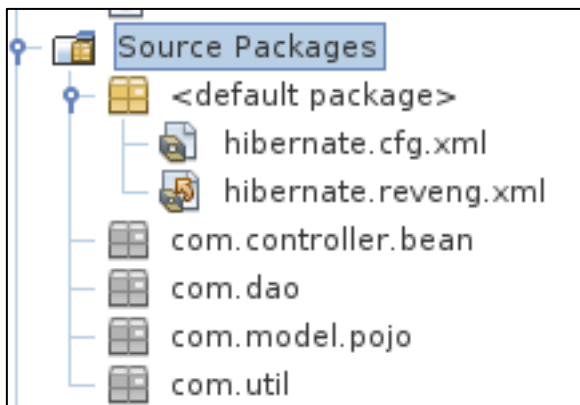
Tras estas opciones pulsamos Finish.

5. Añadimos el driver del conector Mysql



6. Se crean los packages y sus clases

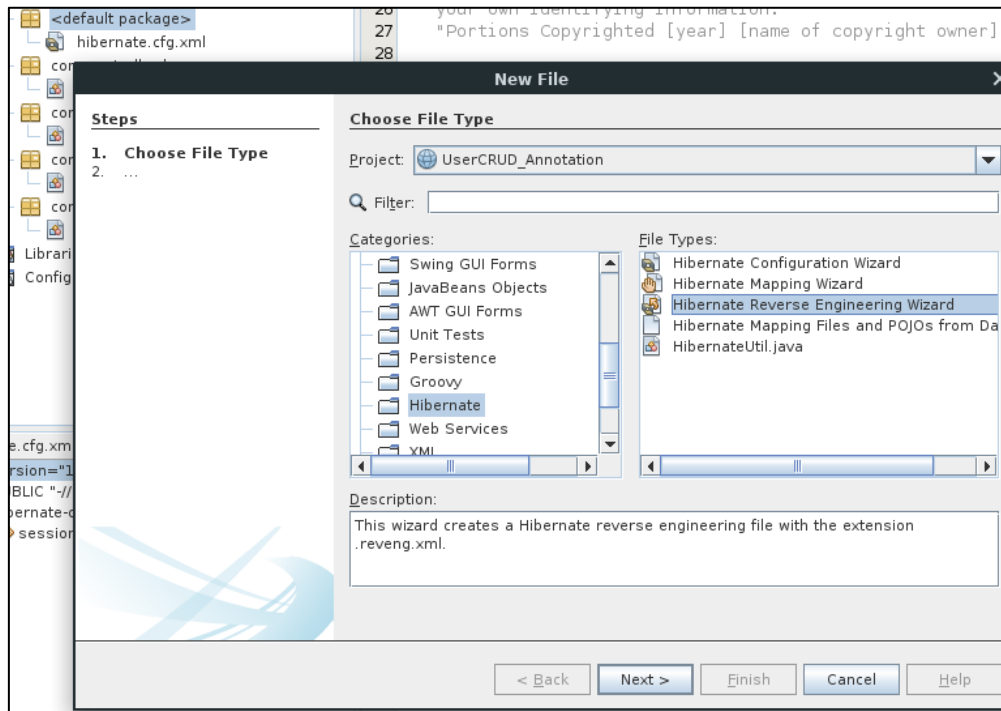
La única clase que se genera automáticamente es hibernate.cfg.xml, el resto hay que crearlas manualmente.



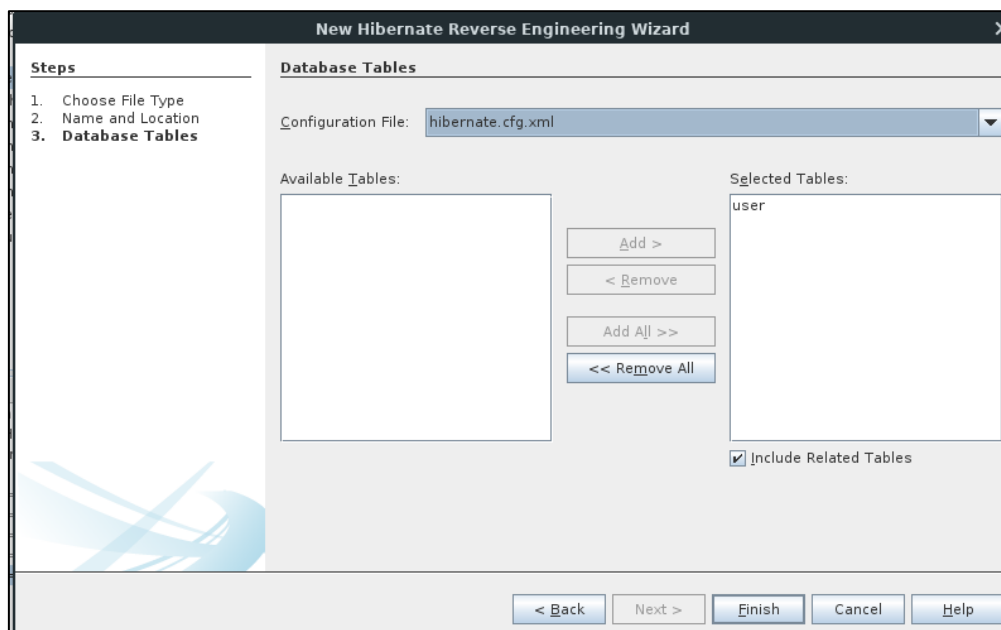
6.1. Creating Reverse Engineering File -> hibernate.reveng.xml

El uso de este fichero es recomendable ya que nos permite regenerar las clases sin perder los cambios.

Para crear el archivo, pulsamos botón derecho sobre “default package”, categories -> Hibernate y en File Types -> Hibernate Reverse Engineering Wizard.



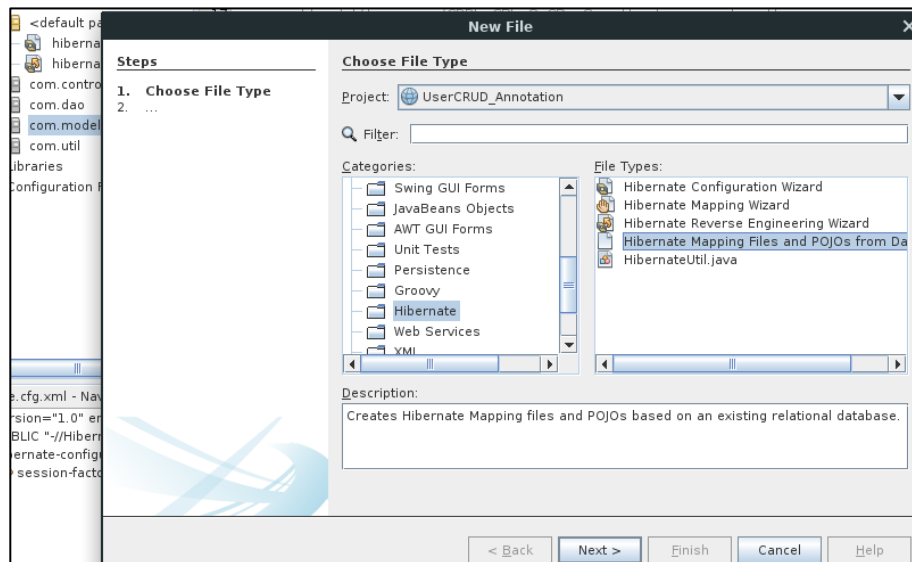
Seleccionamos la tabla user.



6.2. Creating Hibernate Entity (pojo) File: user.java

Esta clase contiene los constructores y los métodos set() y get() para controlar los objetos y permitir leer los datos o modificarlos.

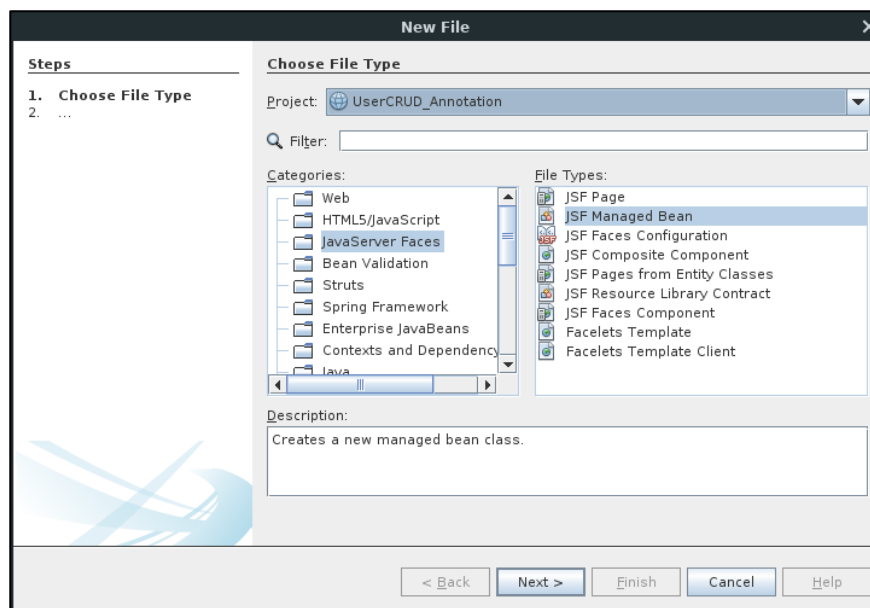
Hacemos click derecho sobre el package com.model y creamos un nuevo archivo Hibernate de tipo Hibernate Mapping Files and POJOs from Database, tras esto le damos a Next y en la siguiente ventana a Finish.



6.3. Creating JSF Managed Bean File UserBean.java

Esta clase contiene los métodos CRUD, que permiten insertar, buscar, modificar y eliminar registros de la tabla.

Para crear este archivo hacemos click derecho en el package com.controller.bean y seleccionamos nuevo archivo, categoría JavaServer Faces, tipo JSF Managed Bean, damos a siguiente, introducimos el nombre y pulsamos Finish.



6.4. Creating DataAccessObject (DAO) File UserDAO.java

Cada DAO realiza las interacciones necesarias con Hibernate para poder actuar sobre un almacen de datos dado. El interface UserDao define los métodos que cada DAO debe implementar.

Hacemos click derecho en el package com.dao y seleccionamos Java Class.

```
public class UserDao
{
    private User user;
    private User newUser;
    private List < User > DaoAllUsers;
    private List < User > DaoSearchUserList;
    //Session session;
    public List < User > AllUsers()
    {
        Session session = HibernateUtil.getSessionFactory().openSe
        try
        {
            session.beginTransaction();
            DaoAllUsers = session.createCriteria(User.class).list()
            int count = DaoAllUsers.size();
            // FacesMessage message1 = new FacesMessage(FacesMessa
            //RequestContext.getCurrentInstance().showMessageInDia
            session.getTransaction().commit();
        }
        catch (Exception e)
        {
            e.printStackTrace();
            session.getTransaction().rollback();
        }
        session.close();
        return DaoAllUsers;
    }
}
```

6.5. HibernateUtil.java

Cuando la clase hibernate.cfg.xml ha analizado sintácticamente todos los mapeos, la aplicación tiene que obtener una fábrica para las instancias org.hibernate.Session. Esta fábrica está concebida para que todos los hilos de la aplicación la compartan.

Click derecho en el package com.util y seleccionamos Java Class.

```
public class HibernateUtil
{
    //Annotation based configuration
    private static SessionFactory sessionFactory;
    private static SessionFactory buildSessionFactory()
    {
        try
        {
            // Create the SessionFactory from hibernate.cfg.xml
            Configuration configuration = new Configuration();
            configuration.configure("/hibernate.cfg.xml");
            System.out.println("Hibernate Annotation Configuration");
            ServiceRegistry serviceRegistry = new StandardServiceRe
            System.out.println("Hibernate Annotation serviceRegistr
            sessionFactory = configuration.buildSessionFactory(serv
            return sessionFactory;
        }
        catch (Throwable ex)
        {
            // Make sure you log the exception, as it might be swa
            System.err.println("Initial SessionFactory creation fai
            throw new ExceptionInInitializerError(ex);
        }
    }
    public static SessionFactory getSessionFactory()
    {
        if (sessionFactory == null) sessionFactory = buildSessionFa
        return sessionFactory;
    }
}
```

6.6. Creating index.xhtml

Esta clase Xhtml configura la página inicial donde veremos la tabla.

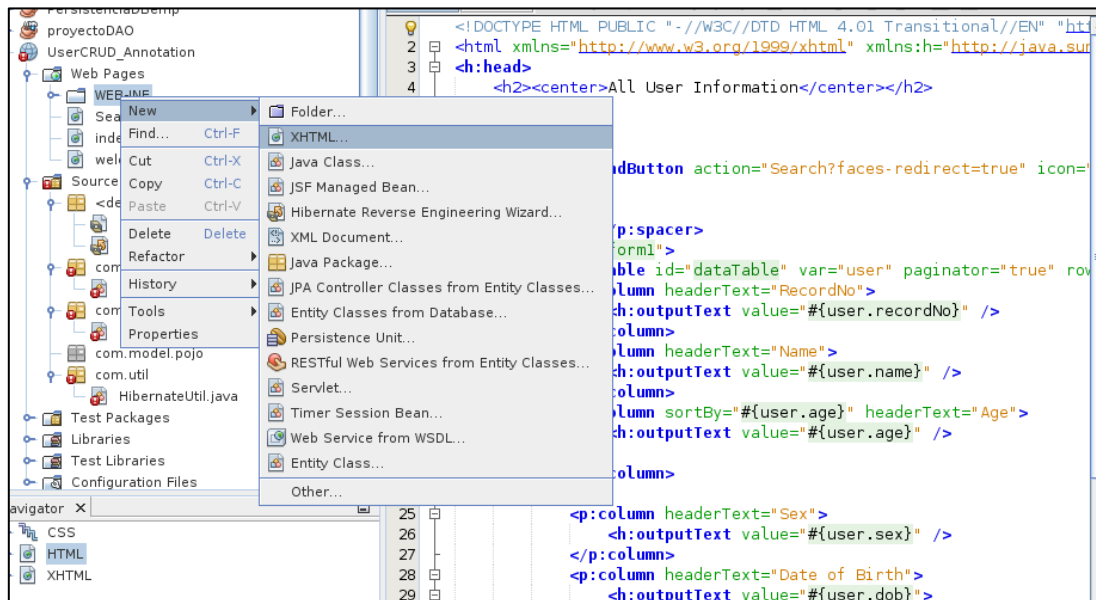
El archivo viene creado por defecto en la carpeta Web Pages, copiamos el texto de la practica guiada y modificamos el que viene.

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html401/
2 <html xmlns="http://www.w3.org/1999/xhtml" xmlns:h="http://java.sun.com/jsf/html"
3 <h:head>
4     <h2><center>All User Information</center></h2>
5 </h:head>
6 <h:body>
7     <h:form>
8         <p:commandButton action="Search?faces-redirect=true" icon="fa-search" />
9     </h:form>
10
11     <p:spacer> </p:spacer>
12     <h:form id="form1">
13         <p:dataTable id="dataTable" var="user" paginator="true" rows="10">
14             <p:column headerText="RecordNo">
15                 <h:outputText value="#{user.recordNo}" />
16             </p:column>
17             <p:column headerText="Name">
18                 <h:outputText value="#{user.name}" />
19             </p:column>
20             <p:column sortBy="#{user.age}" headerText="Age">
21                 <h:outputText value="#{user.age}" />
22             </p:column>
23
24             <p:column headerText="Sex">
25                 <h:outputText value="#{user.sex}" />
26             </p:column>
27             <p:column headerText="Date of Birth">
28                 <h:outputText value="#{user.dob}" />
29                 <f:convertDateTime type="date" pattern="dd-MMM-yy" />
30             </p:column>
31             <p:column headerText="Remark">
32                 <h:outputText value="#{user.remark}" />
33             </p:column>
34         </p:dataTable>
35     </h:form>
36     <h:panelGroup>
37         <h3>Add User Information</h3>
38         <h:form>
39             <p>User Name:
40                 <p:inputText value="#{userBean.newuser.name}" /></p>
41         </h:form>
42     </h:panelGroup>
43 </h:body>
44 </html>
```

6.7. Search.xhtml

Esta clase Xhtml configura la página de búsqueda donde tras seleccionar el botón Search en la página index.xhtml no redirigirá a esta donde podremos buscar user por primary key y también permite modificar y eliminar registros.

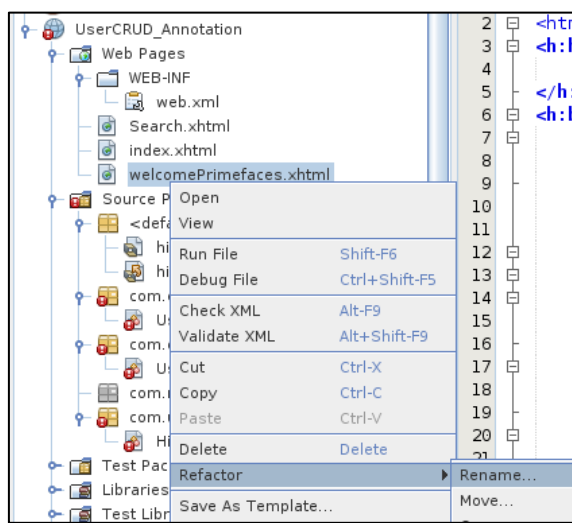
Para crear este archivo en la carpeta Web Pages hacemos click derecho sobre WEB-INF y seleccionamos New XHTML.



6.8. UserCRUD.xhtml

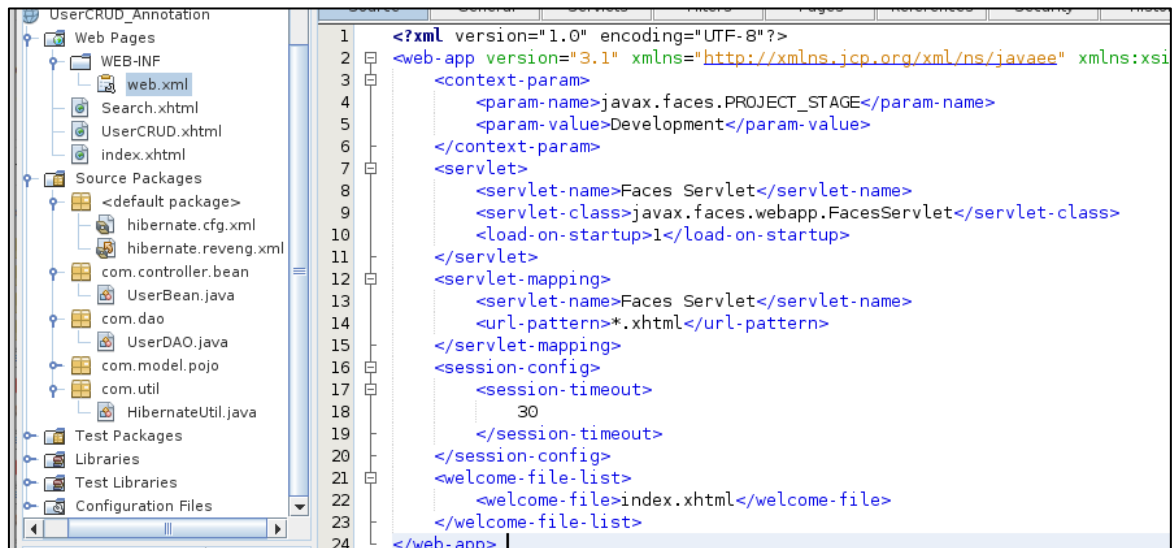
Esta clase es una web que se puede usar de forma independiente, en ella se ver una web para usar los métodos CRUD.

Este archivo ya viene creado por defecto con otro nombre welcomePrimefaces.xhtml, hacemos refactor y cambiamos el nombre por UserCRUD.



6.9. web.xml (Automatically generated)

El archivo ya viene generado automáticamente y no hay que modificar el código.



7. Se ejecuta el programa

File Edit View History Bookmarks Tools Help

localhost:8080/UserCRUD_ x +

localhost:8080/UserCRUD_Annotation2/ ☆ Q Search

Help Manual Support Forums Google Search

All User Information

Search

RecordNo	Name	Age	Sex	Date of Birth	Remark
1	Raichand	48	Male	16-Jun-2014	Good Person
2	Ramesh	45	Male	18-Mar-2013	Good Person
3	smitta	47	Female	09-Jan-1996	Jolly Lady
4	Sarita	37	Female	07-Jan-1998	good beautiful lady
7	Rashmi	23	Male	25-Jan-2016	good person

Add User Information

User Name:

User Age: Enter Number Only

Choose Sex:

User Date of Birth:

Remark:

Activar Windows

7.1. Insertar User

Debajo de la tabla hay un formulario para añadir un usuario nuevo, pulsamos en el botón Add.

Add User Information

User Name:

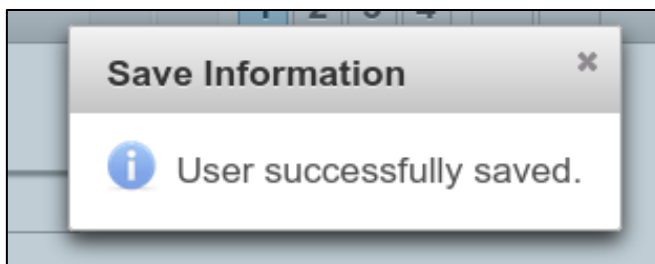
User Age: Enter Number Only

Choose Sex:

User Date of Birth:

Remark:

Si se han introducido los datos correctamente saldrá un aviso indicando que se ha insertado en la tabla



7.2. Buscar User

Pulsamos el botón Search y nos llevara a una nueva ventana, seleccionamos el numero de registro que queremos buscar, y volvemos a pulsar search y se muestran los datos obtenidos de ese usuario.

Search User For Edit And Delete

Please Select Record Number:

Row Editing							
RecordNo	Name	Age	Date of Birth	Sex	Remark	Edit	Delete
29	Jorge	34	30-Sep-1988	Male	Experto en bases de datos		<input type="button" value="Delete"/>

7.3. Modificar User

Para modificar un usuario, lo buscamos como en el punto anterior y pulsamos sobre el icono de lapicero y editamos, tras hacer los cambios, pulsamos sobre el icono check para confirmar o sobre la X para cancelar la modificación.

Search User For Edit And Delete

← Back

Please Select Record Number:

Search

Row Editing							
RecordNo	Name	Age	Date of Birth	Sex	Remark	Edit	Delete
29	Carlos	34	30/09/1988	Ma	Experto en ba	✓ ✕	Delete

Cuando guardemos los datos se mostrará un aviso indicando la modificación.

Search User For Edit And Delete

← Back

Please Select Record Number:

Search

i Edited Record No
29

Row Editing							
RecordNo	Name	Age	Date of Birth	Sex	Remark	Edit	Delete
29	Carlos	34	30-Sep-1988	Male	Experto en bases de datos	✎	Delete

7.4. Borrar User

Por último, para borrar un usuario pulsamos sobre el botón delete, nos saldrá un aviso de si queremos confirmar la eliminación o no.

Row Editing							
RecordNo	Name	Age	Date of Birth	Sex	Remark	Edit	Delete
29							Delete

Deleting ✕

⚠ Are you sure you want to delete this record?Record once deleted can not be retrieved.

Yes Sure Not Yet

Tras pulsar la confirmación veremos una notificación.

Delete ✕

i Record deleted successfully