

## Soluciones Ejercicios propuestos Framework Hibernate:

1.-) El código del método comprobar() del bean LoginBean sería el siguiente:

```
public String comprobar() {  
  
    if (nombre.equals("fulanito")) return "error";  
    Session session = HibernateUtil.getSessionFactory().getCurrentSession();  
    session.beginTransaction();  
  
    String preg="from Usuario where nombre='"  
        +nombre+"' and password='"+password+"' and tipo='"+tipo.toString()+"";  
  
    List result = session.createQuery(preg).list();  
    int num = result.size();  
  
    session.getTransaction().commit();  
    if (num==0) return "noexiste";  
    else return "ok";  
}
```

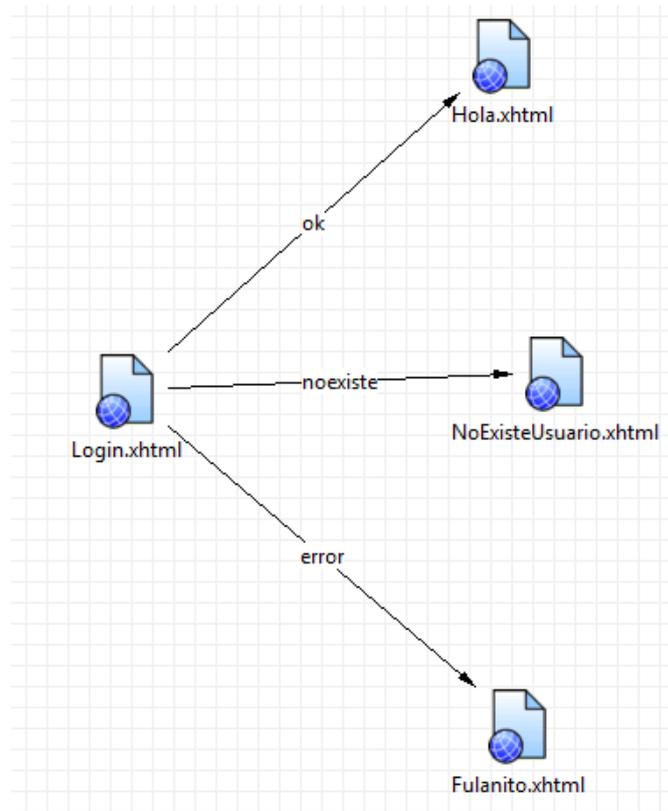
Además hay que crear un nuevo JSF, llamado por ejemplo NoExisteUsuario.xhtml, para el caso en el que el usuario no se ha identificado correctamente. En dicho JSF se muestra, con `<h:outputText value="#{Login.nombre}"/>`, el nombre con el que se ha intentado identificar:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
  
<html xmlns="http://www.w3.org/1999/xhtml"  
    xmlns:ui="http://java.sun.com/jsf/facelets"  
    xmlns:h="http://java.sun.com/jsf/html"  
    xmlns:f="http://java.sun.com/jsf/core">  
  
    <f:view>  
    <h:head>  
        <title>Una simple aplicacion JavaServer Faces</title>  
    </h:head>  
    <h:body>  
    <h:form>  
    <h3> <h:outputText value="#{Login.nombre}"/>  
    no tiene permiso para entrar al sistema</h3>  
  
    </h:form>  
    </h:body>  
    </f:view>  
</html>
```

Por último, hay que modificar el faces-config.xml para que se navegue correctamente de una vista a otra cuando el método action comprobar lanza la acción "noexiste"

```
<navigation-rule>  
    <display-name>Login.xhtml</display-name>  
    <from-view-id>/Login.xhtml</from-view-id>  
    <navigation-case>  
        <from-outcome>noexiste</from-outcome>  
        <to-view-id>/NoExisteUsuario.xhtml</to-view-id>  
    </navigation-case>  
</navigation-rule>
```

Esa regla de navegación se puede escribir o editar de manera gráfica:



2.-) El código del método `comprobar()` del bean `LoginBean` en este caso no contendría código Hibernate, sino que delegaría la comprobación de si el usuario existe en una clase de acceso a datos: `UsuarioDAO` en este caso.

```
public String comprobar() {  
    if (nombre.equals("fulanito")) return "error";  
    if (UsuarioDAO.hacerLogin(nombre, password, tipo)) return "ok";  
    else return "noexiste";  
}
```

La clase de acceso a datos sería la que implementaría el método `hacerLogin`, que haría prácticamente lo mismo que hacía `comprobar()` de `LoginBean` en el ejercicio anterior. Además, a esa clase se le ha añadido el método `createAndStoreUsuario` de `CrearEventos`, porque como se ha dicho en el enunciado esa clase tenía métodos típicos de clases DAO (aunque se han cambiado `private` de a `public static`). Se presenta a continuación la implementación de la clase `UsuarioDAO`

```
package modelo;

import java.util.List;

import org.hibernate.Session;

import modelo.dominio.Usuario;
import principal.CrearEventos;

public class UsuarioDAO {

    public static void createAndStoreUsuario(String nombre, String password, String tipo){
        Session session = HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();

        Usuario u = new Usuario();
        u.setNombre(nombre);
        u.setPassword(password);
        u.setTipo(tipo);
        session.save(u);

        session.getTransaction().commit();
    }

    public static boolean hacerLogin(String nombre, String password, String tipo){
        Session session = HibernateUtil.getSessionFactory().getCurrentSession();
        session.beginTransaction();

        String preg="from Usuario where nombre='"
            +nombre+"' and password='"+password+"' and tipo='"+tipo.toString()+"'";

        List result = session.createQuery(preg).list();
        int num = result.size();

        session.getTransaction().commit();

        return (num>0);
    }

    public static void main(String[] args) {
        UsuarioDAO e = new UsuarioDAO();
        System.out.println("Creación de usuarios:");

        e.createAndStoreUsuario("Pepe", "125", "estudiante");
        e.createAndStoreUsuario("Nerea", "125", "estudiante");
    }
}
```

