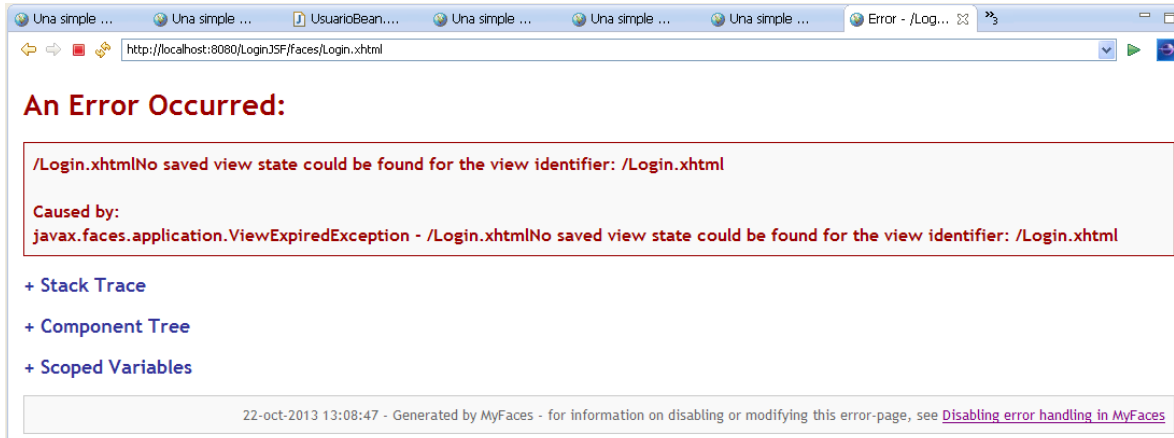


Solucionando algunos errores

1) El servidor web no carga los JSFs más recientes

A veces el servidor web no muestra correctamente los últimos JSFs de la aplicación web y se muestran mensajes como los siguientes:



Una simple ... Una simple ... UsuarioBean... Una simple ... Una simple ... Una simple ... Error - /Log...
http://localhost:8080/LoginJSF/faces/Login.xhtml

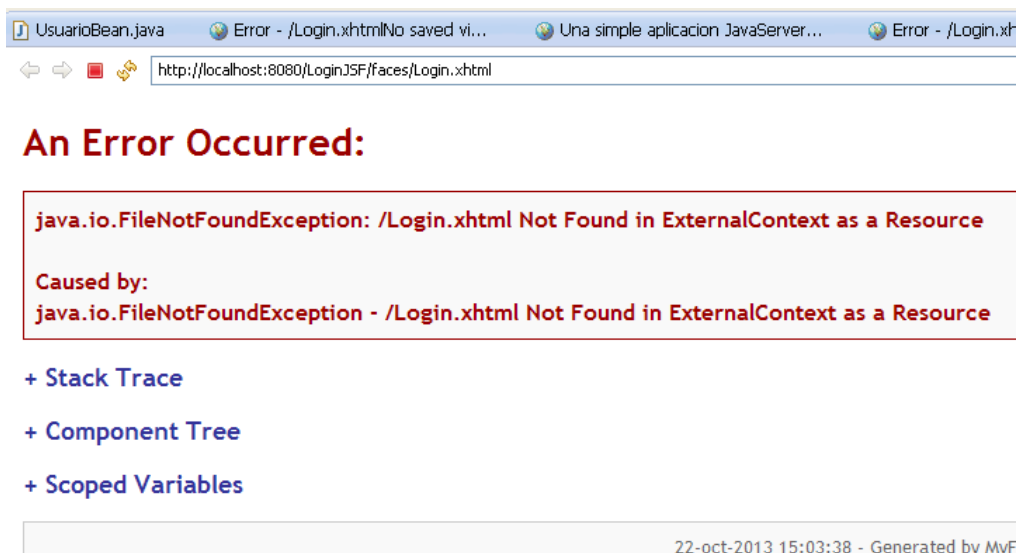
An Error Occurred:

/Login.xhtmlNo saved view state could be found for the view identifier: /Login.xhtml

Caused by:
javax.faces.application.ViewExpiredException - /Login.xhtmlNo saved view state could be found for the view identifier: /Login.xhtml

+ Stack Trace
+ Component Tree
+ Scoped Variables

22-oct-2013 13:08:47 - Generated by MyFaces - for information on disabling or modifying this error-page, see [Disabling error handling in MyFaces](#)



UsuarioBean.java Error - /Login.xhtmlNo saved vi... Una simple aplicacion JavaServer... Error - /Login.xf
http://localhost:8080/LoginJSF/faces/Login.xhtml

An Error Occurred:

java.io.FileNotFoundException: /Login.xhtml Not Found in ExternalContext as a Resource

Caused by:
java.io.FileNotFoundException - /Login.xhtml Not Found in ExternalContext as a Resource

+ Stack Trace
+ Component Tree
+ Scoped Variables

22-oct-2013 15:03:38 - Generated by MyF



UsuarioBean... Una simple ... Una simple ... Una simple ... Error - /Log
http://localhost:8080/LoginJSF/faces/Login.xhtml

Estado HTTP 404 - /LoginJSF/faces/Login.xhtml

type Informe de estado

mensaje /LoginJSF/faces/Login.xhtml

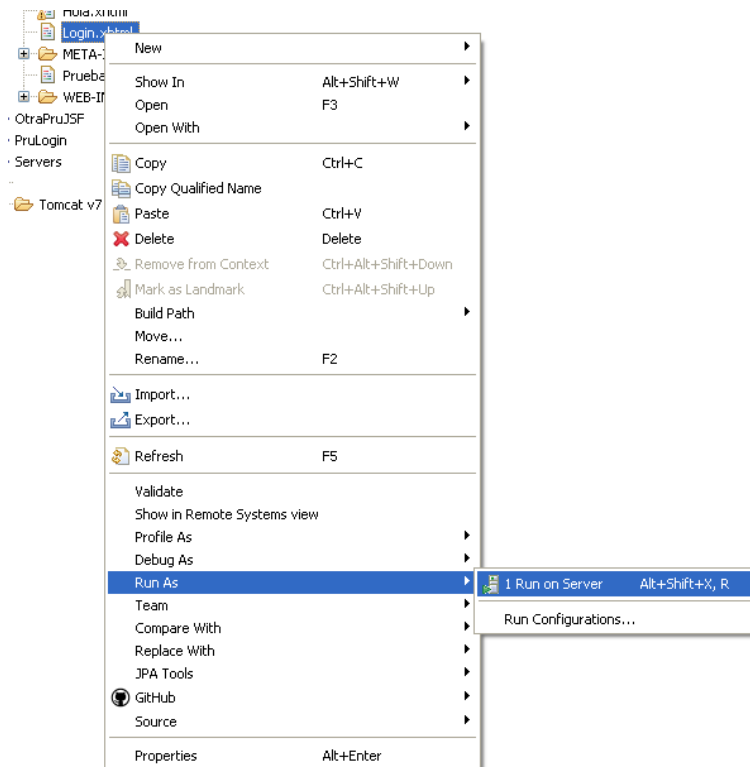
descripción El recurso requerido (/LoginJSF/faces/Login.xhtml) no está disponible.

Apache Tomcat/7.0.21

En estos casos se puede intentar hacer lo siguiente:

- **Recargar el JSF en el navegador para que se traiga la versión del servidor y no la que pueda haber en la cache.**

Hacer click derecho sobre el JSF y escogiendo Run As => Run on Server

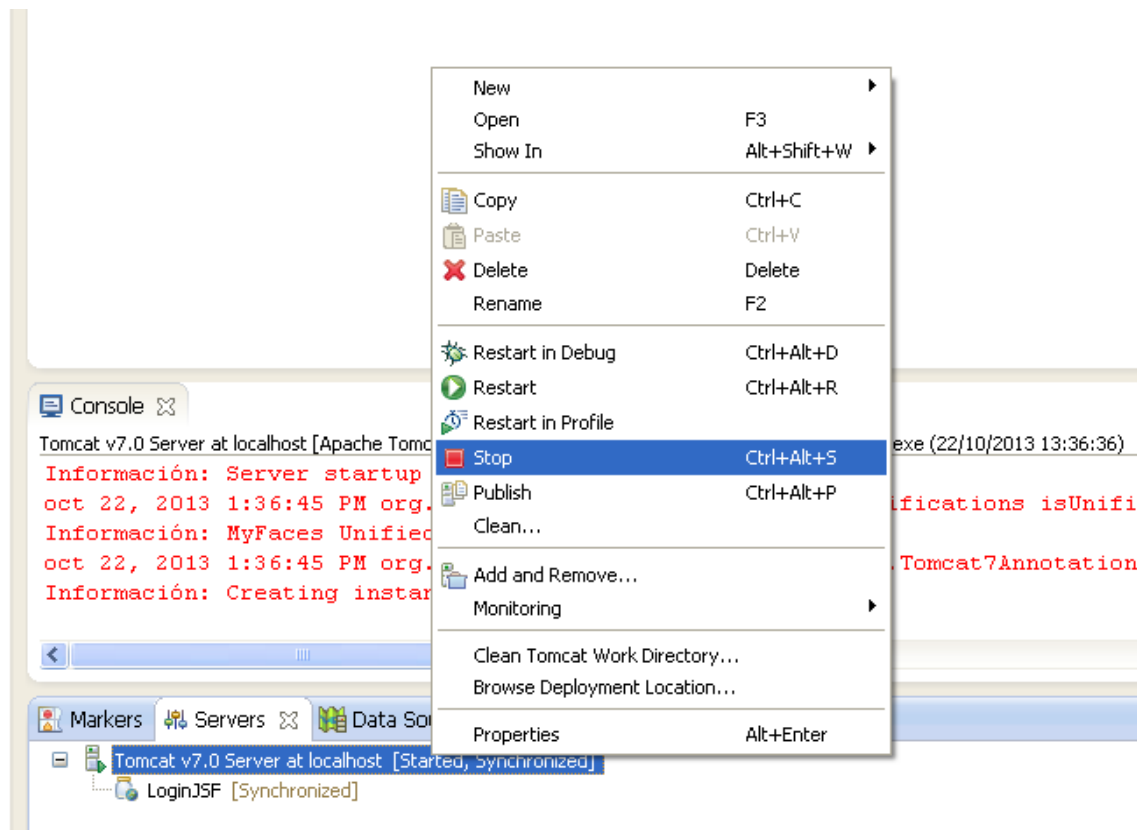


Y refrescar el JSF en el navegador pulsando sobre el triángulo verde (ver flecha roja) en el navegador embebido en Eclipse (o en el navegador donde se haya invocado a la aplicación web).

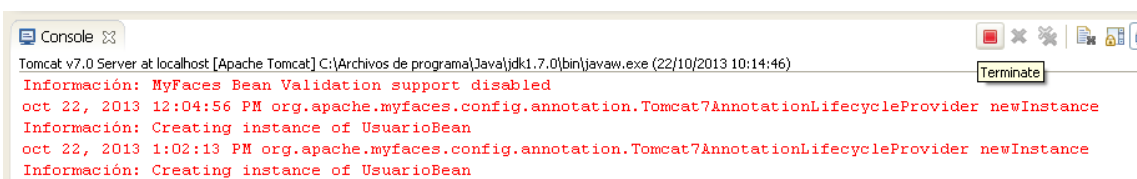


- **Reiniciar el servidor web antes de ejecutar el JSF**

Si recargando el JSF no se carga, se puede proceder a terminar/reiniciar el servidor web (con las opciones Stop o Restart) que aparecen al hacer click derecho sobre el servidor web en la pestaña Servers.

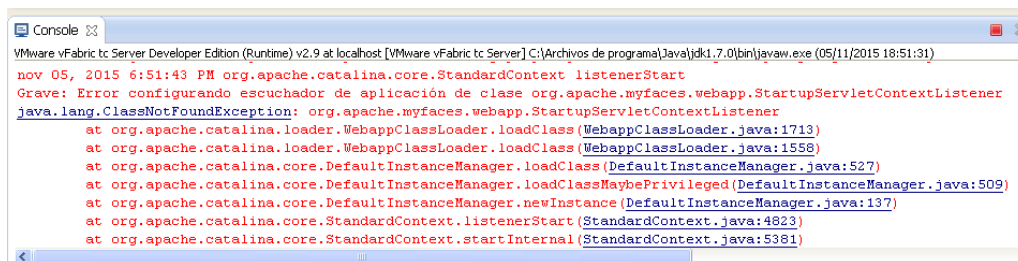


Pulsando el icono “Terminate” o con las cruces que aparecen al lado en la consola.



2) El servidor web no encuentra las librerías

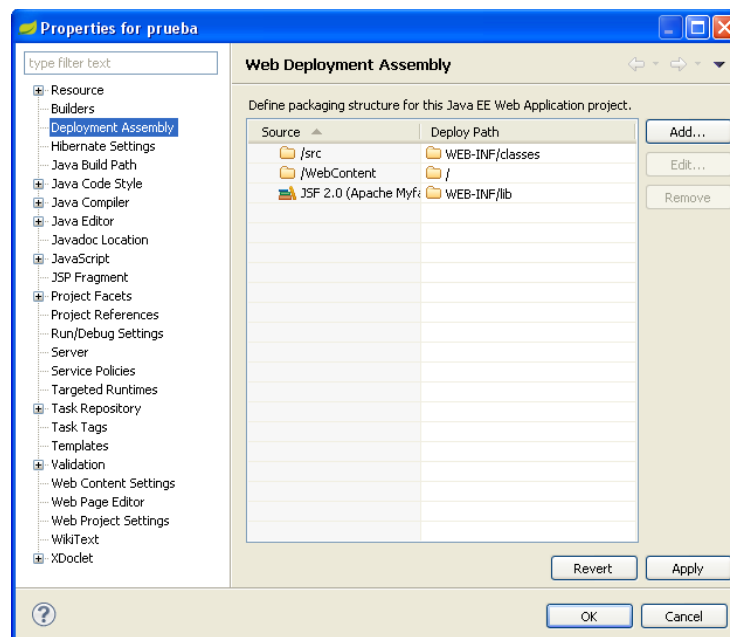
A veces el servidor web no encuentra alguna clase perteneciente a alguna de las librerías usadas en la aplicación web, aunque dichas clases sí estén disponibles en el proyecto Eclipse. Por ejemplo, puede suceder que no encuentre una clase como WebappClassLoader de myfaces (una de las implementaciones de JSF):



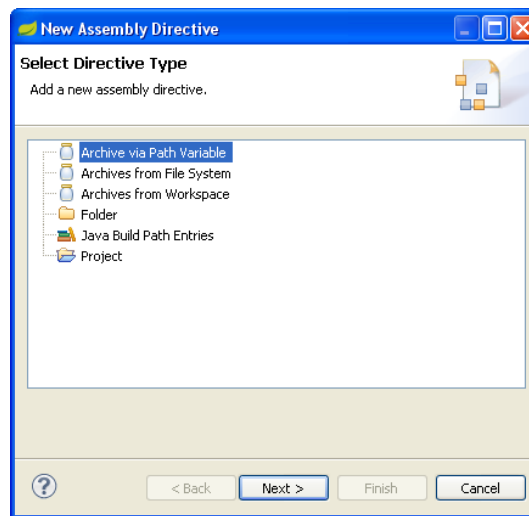
En este caso hay que hacer lo siguiente:

- **Desplegar todas las librerías correctamente en el servidor Web**

Para ello: Project => Properties => Deployment Assembly



=> Add



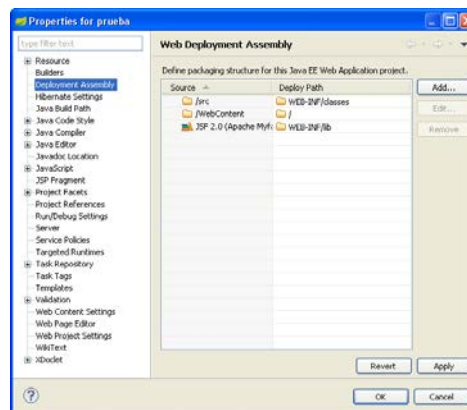
=> Java Build Path Entries => Next => Seleccionar los .jar que se necesiten desplegar (deploy) en el directorio lib de WEB-INF para que estén disponibles para el servidor web.

3) El servidor web no encuentra los beans de la aplicación web

Otras veces no se encuentran las clases bean del proyecto, o no utiliza la última versión del bean tras haberlo actualizado en el proyecto. En este caso hay que asegurarse como antes de:

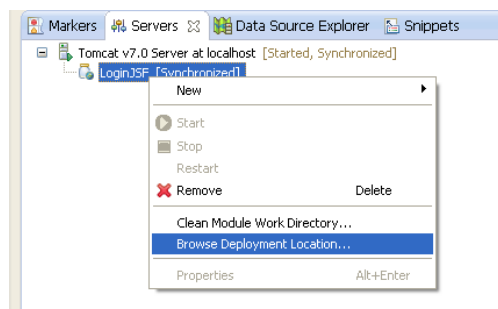
- **Desplegar todas las clases del proyecto**

Para ello: Project => Properties => Deployment Assembly => Asegurarse de que las clases del directorio src se despliegan en el directorio WEB-INF/classes del servidor web.

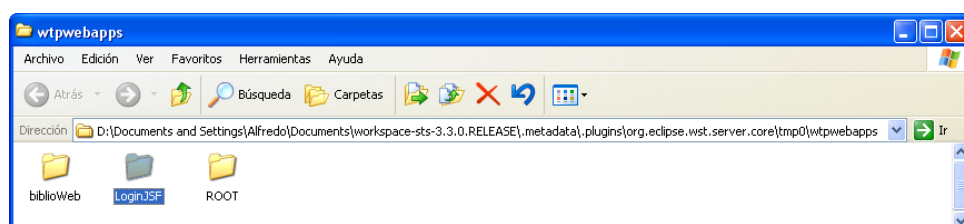


- **Comprobar que el bean se despliega correctamente en el servidor web**

Se puede ver exactamente qué JSFs, beans, etc. son los que están ejecutándose en el servidor web en ese momento, y si están desplegadas las librerías correctamente. Haciendo "Browse Deployment Location" como se indica aquí:

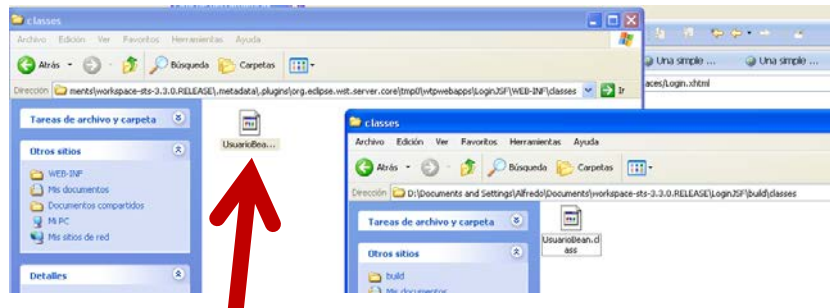


Y navegando por los contenidos de la aplicación Web (por ejemplo LoginJSF), en concreto por la carpeta WEB-INF y dentro de ella, por clases y lib, a ver si están los .jar y si el .class del bean es el correcto (mirando la fecha y hora, por ejemplo).



Si no estuvieran, entonces hay que asegurarse de que se compila el proyecto (con Build Project) y que al ejecutarlo se copian automáticamente bien las clases ahí. A veces Eclipse no recompila porque no detecta correctamente que la clase ha cambiado. Se puede modificar y volver a salvar la clase para obligar a que Eclipse la recompile.

Si la clase desplegada no es la última, se podría copiar y pegar a mano en las carpetas del servidor web. Los BEAN irían en el subdirectorio “clases” de WEB-INF, las librerías en el subdirectorio “lib” de WEB-INF y los JSFs en la misma carpeta del proyecto desplegado (en este caso LoginJSF)



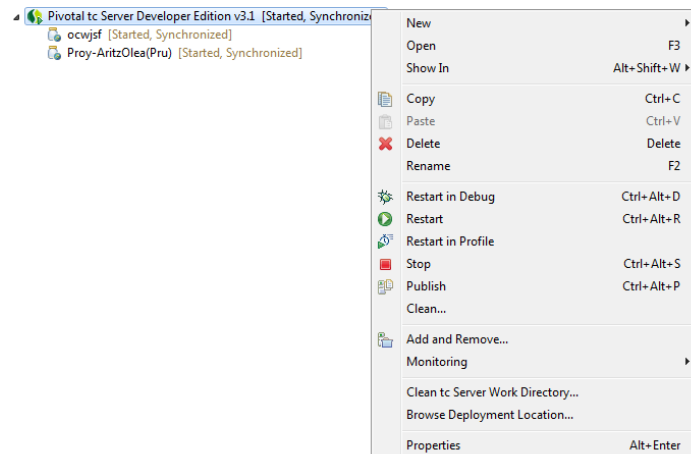
**Copiada “a mano” la
clase desplegada en el
servidor web**

También es posible que el proyecto esté en un estado incorrecto y puede convenir limpiar el proyecto y/o el directorio del servidor web donde se despliega la aplicación web.

Limpiar el proyecto: Project => Clean

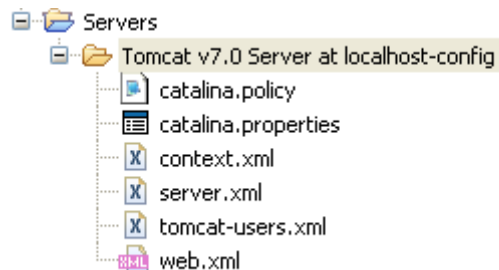
O bien:

Limpiar el servidor: Click dcho sobre servidor Web => Clean tc Server Work Directory



4) El puerto en el que se lanza el servidor web está ocupado

Se puede cambiar el puerto por defecto (el 8080) que se encuentra en el fichero server.xml, teniendo en cuenta que escogemos el servidor que realmente se vaya a ejecutar.



```
<Connector connectionTimeout="20000" port="8080" protocol="HTTP/1.1"
redirectPort="8443"/>
```

Cuidado: en el fichero server.xml aparecen otros puertos. No confundirlos con el que atiende al protocolo HTTP.

Por ejemplo: <Server port="8005" shutdown="SHUTDOWN">

Además, este último es el que más fácilmente se ve (flecha roja) en vez del anterior (flecha verde), que puede estar oculto si no está desplegado el nodo del Connector.

