**GRUPO 6**

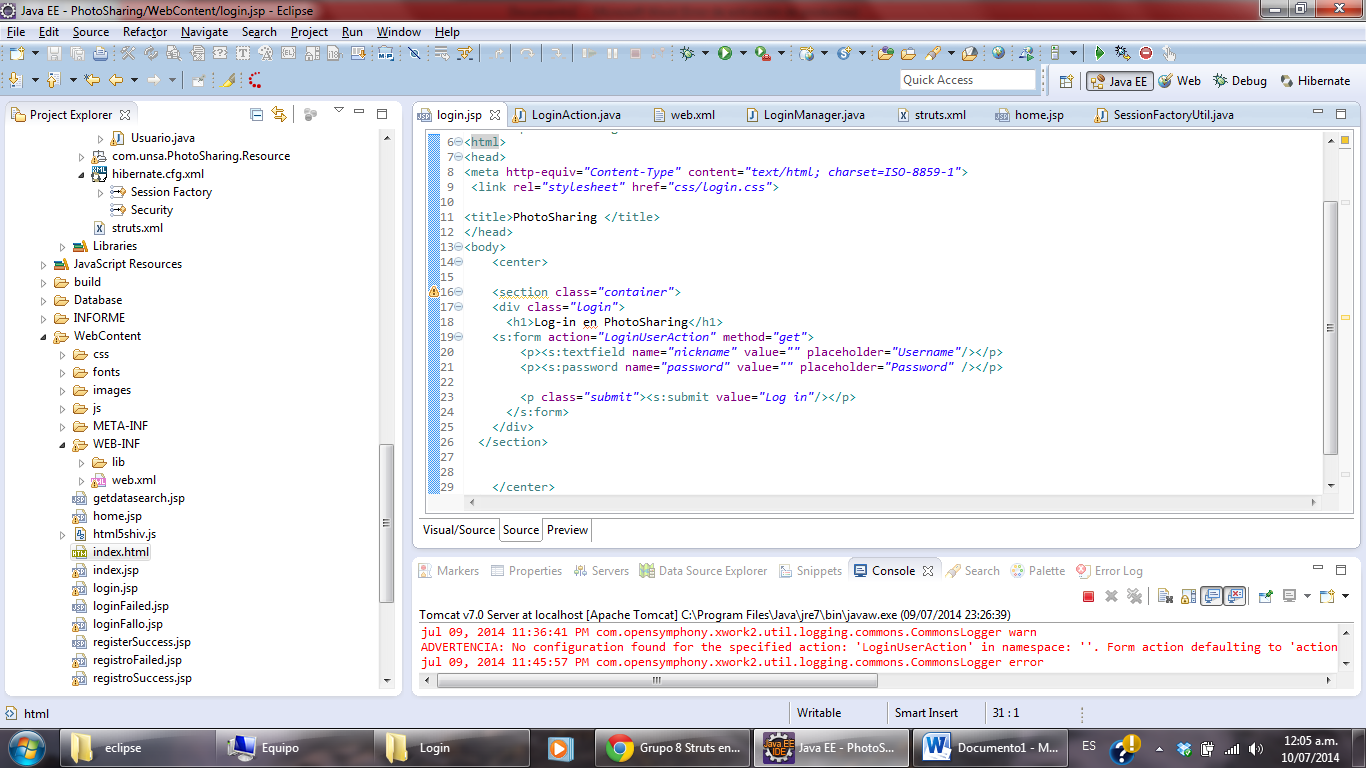
**RED SOCIAL PARA COMPARTIR FOTOS**

**STRUTS**

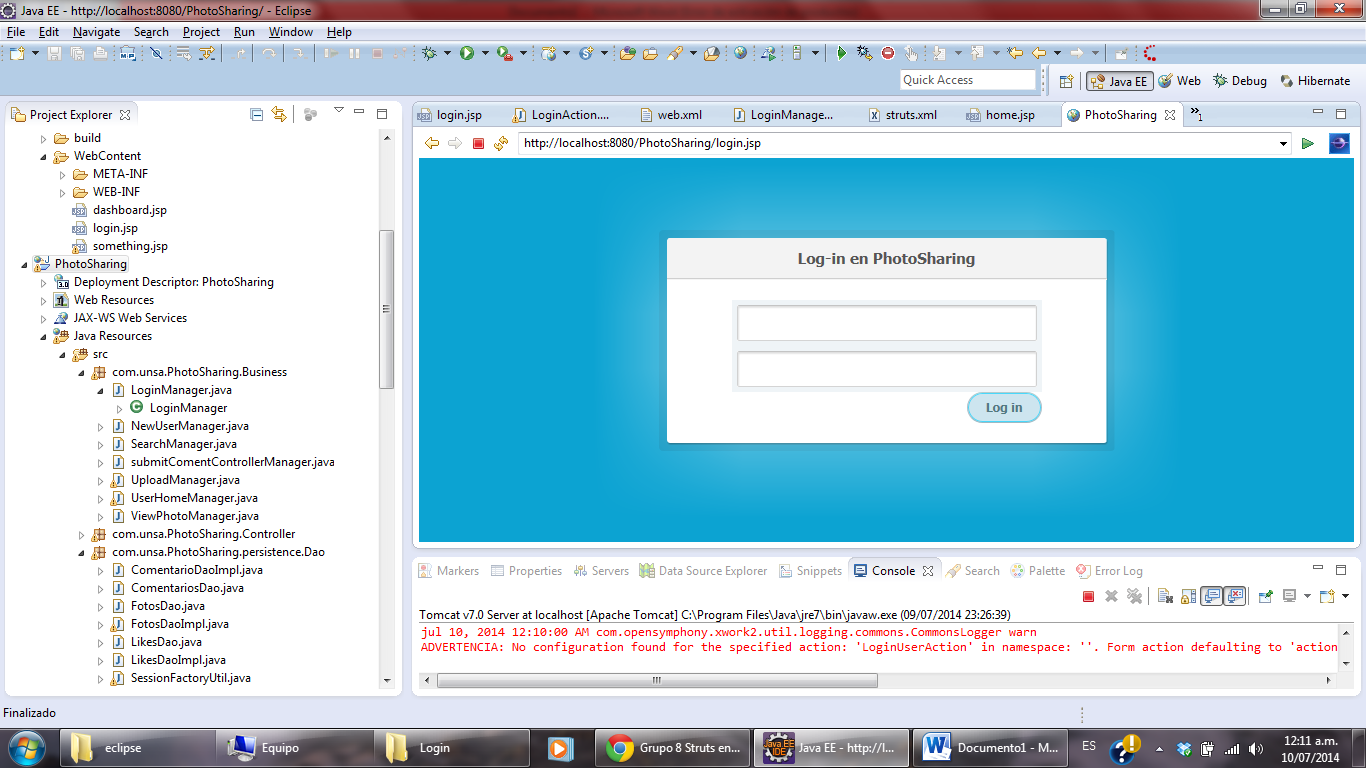
Se empleó Struts en el módulo de Login. El manejo de Actions de Struts pertenece a la capa de Controlador en la arquitectura definida.

Para poder satisfacer los requerimientos, se realizaron los siguientes cambios:

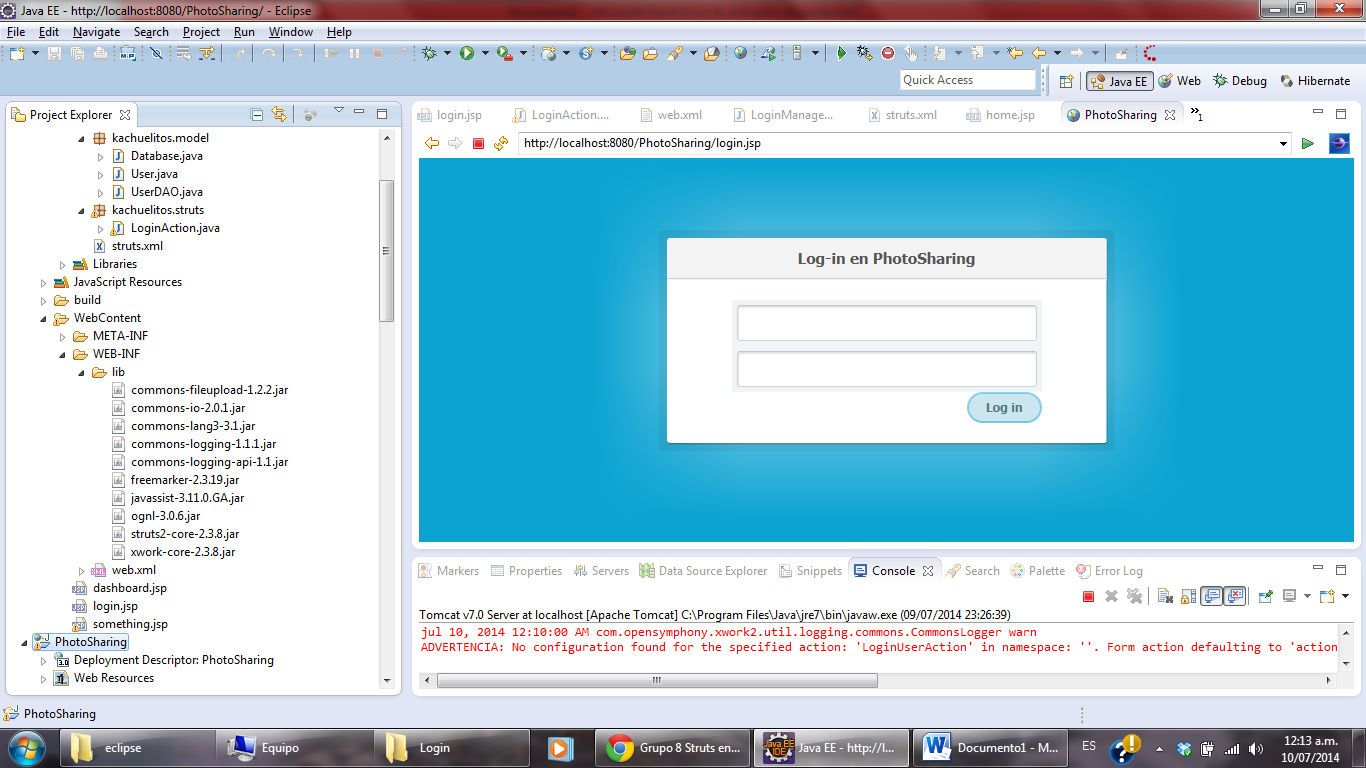
1. Archivo login.jsp con etiquetas propias de Struts.



*Figura 1. Archivo login.jsp*

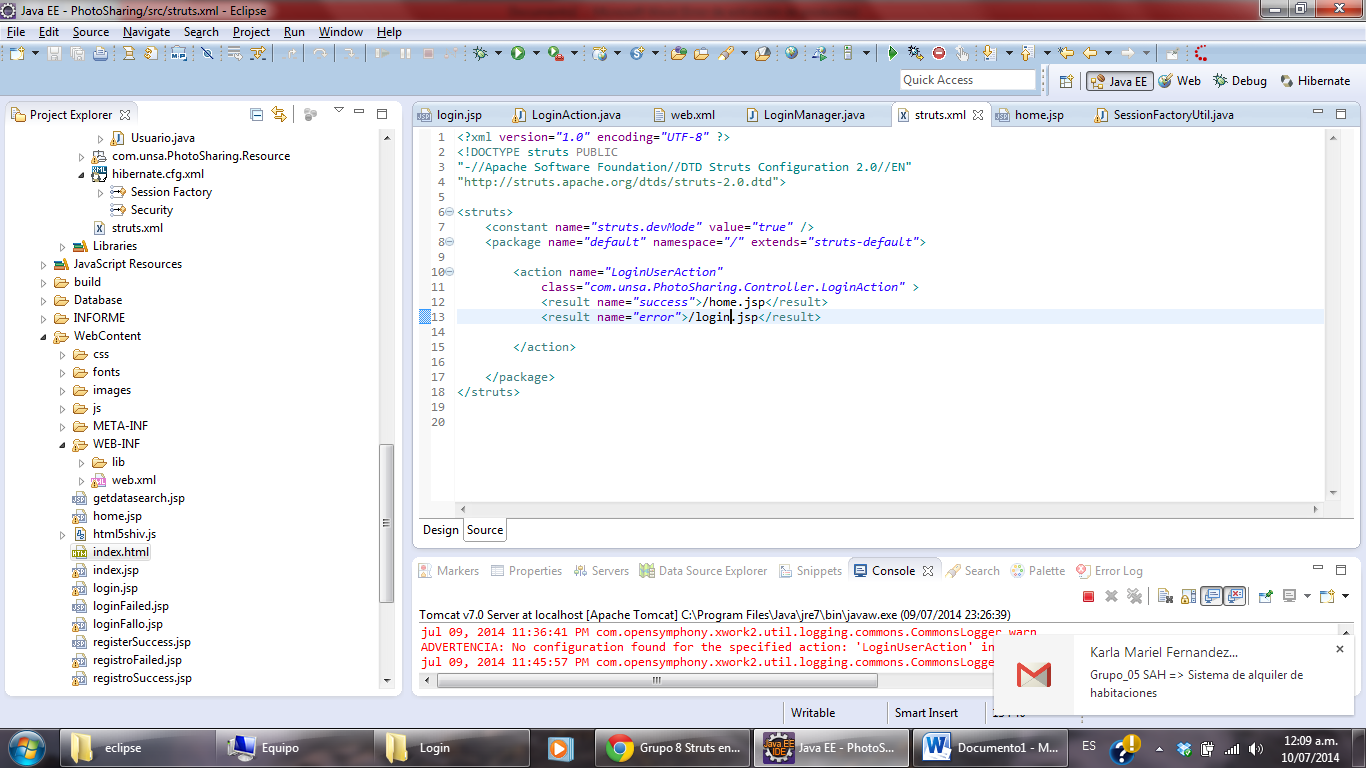


1. Adición de librerías necesarias para el uso del framework. La siguiente lista muestra los archivos \*.jar necesarios para el desarrollo bajo Struts:



*Figura 2. Librerías requeridas*

1. Archivo Struts.xml, con la configuración necesaria para hacer el bindeo de las clases definidas y los action del lado de interfaz.

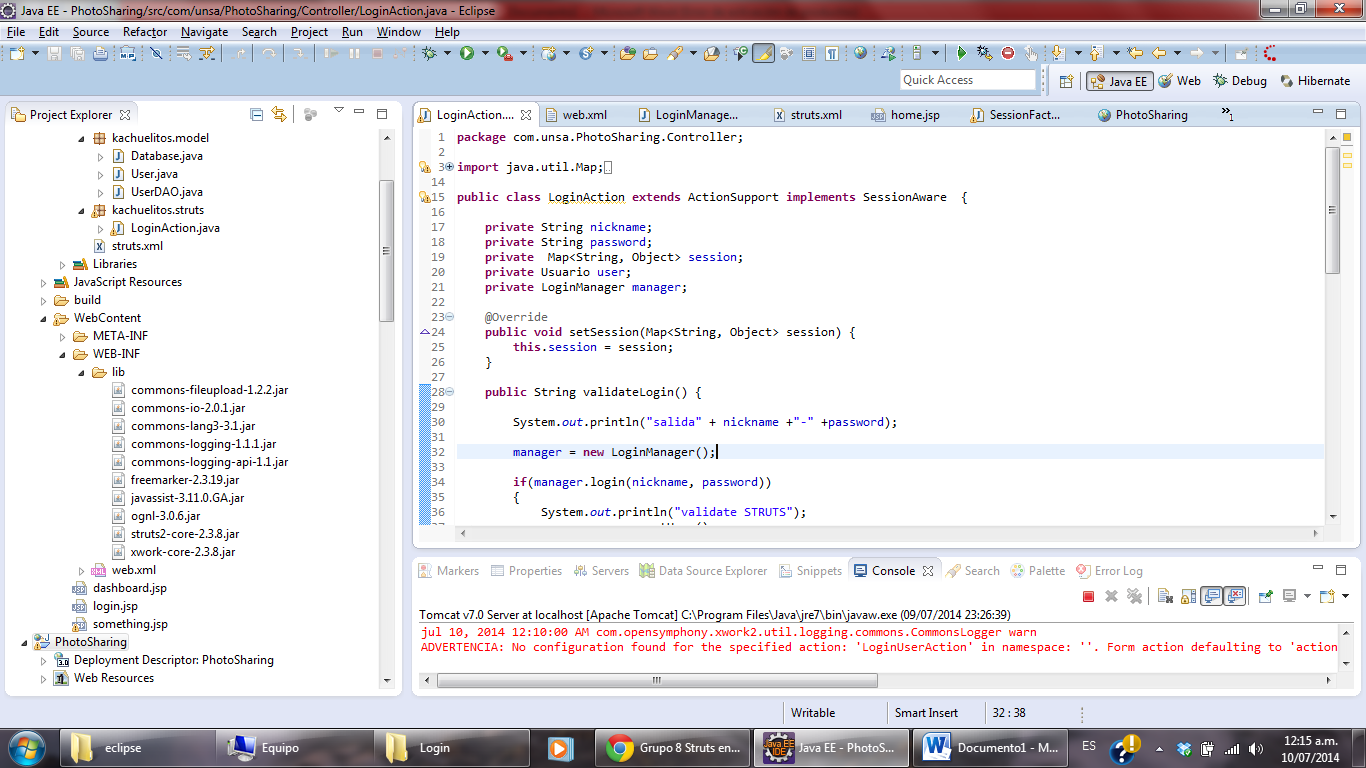


*Figura 3. Archivo Struts.xml*

En el archivo de configuración de Struts, se define un action llamado LoginUserAction, que se ejecutará siempre que el usuario intente realizar un logeo en la aplicación.

La definición de esta clase se encuentra en el paquete PhotoSharing.Controller.LoginAction, y cuando la respuesta al intento de logeo sea exitosa, se maneja una redirección a la página de inicio. Si el intento es fallido, la redirección es hacia la misma página de login, para solicitar un nuevo intento al momento de ingresar los datos.

1. Clases que implementan la interfaz SessionAware (manejo de sesiones) y heredan de la clase ActionSupport (manejo de acciones y bindeo con interfaz por parte de Struts)



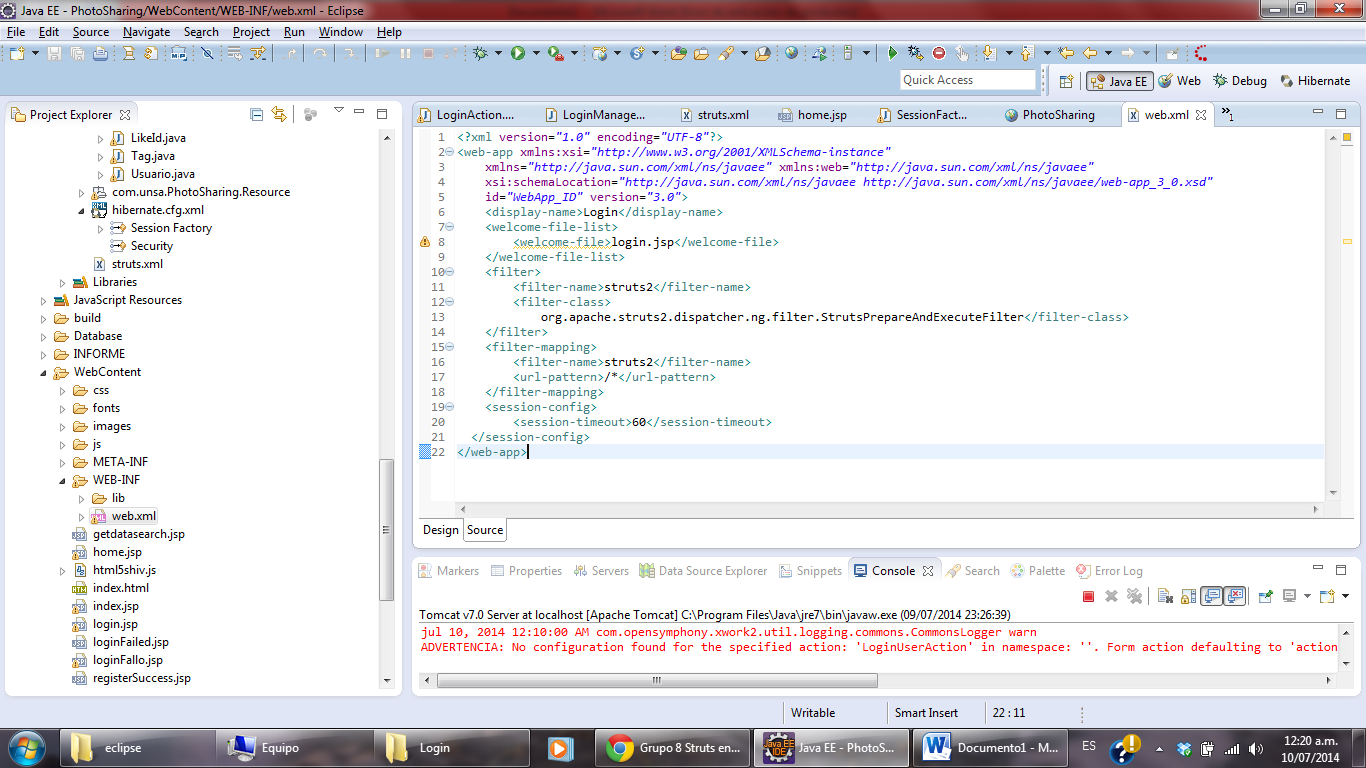
*Figura 4. Prototipo de la clase LoginAction*

La clase LoginAction, al estar bindeada con la vista de login, tiene dos atributos principales: **nickname** y **password**, los mismos que nos servirán para realizar la verificación de credenciales en la base de datos.

El atributo **session** nos permite realizar un manejo de sesiones desde Struts.

Los atributos **user** y **manager** son los que nos permiten interactuar con las otras capas de la aplicación (**manager** para relacionar con la lógica del negocio y hacer la validación de los campos ingresados), y **user** para obtener como objeto, aplicando mapeo objeto-relacional, la información del usuario logueado.

1. Finalmente, el archivo web.xml también sufrió variaciones al momento de agregar los filtros de Struts.



*Figura 5. Archivo web.xml*

**DIVISIÓN DE TAREAS:**

Jan Llerena

Definición de la clase LoginAction

Relación con clases de la arquitectura previamente definida

Manejo de sesiones y validación de credenciales

Jhon Monroy

Diseño de Interfaz empleando tags de Struts

Implementación del archivo login.jsp

Julio Vera

Variaciones en web.xml para añadir filtros de Struts

Elaboración del archivo struts.xml

Definición del mapeo entre vista y Actions. Navegabilidad