

# Desarrollo de Interfaces

## Práctica 2.1

### 1. Formato y entrega

El proyecto se deberá desarrollar con el IDE Eclipse y tendrá el siguiente nombre: **DII\_P2.1\_NOMBRE\_APELLIDO1\_APELLIDO2**, siendo NOMBRE tu nombre propio y APELLIDO1 y APELLIDO2 tus apellidos respectivamente separados por guiones bajos, en mayúsculas y todo sin tildes, ni caracteres especiales, ni eñes, ni espacios (si tu nombre y/o apellidos son compuestos, dejarlos juntos, es decir, sólo deben aparecer 2 guiones bajos).

El paquete base de las clases que formen el proyecto será el siguiente: **dam2.dii.p21** (a partir de ahí puedes montar la estructura que quieras).

La versión del JDK con la que se compilará y entregará el proyecto será la **8u212**, y la **7** en el caso del servidor Tomcat utilizado para el despliegue.

La entrada a la aplicación se hará a través de la url base del proyecto (**[http://localhost:8080/DII\\_P2.1\\_NOMBRE\\_APELLIDO1\\_APELLIDO2](http://localhost:8080/DII_P2.1_NOMBRE_APELLIDO1_APELLIDO2)**) directamente a un fichero **index.jsp**.

Respecto a la nomenclatura a seguir de los elementos que componen el programa, se recomienda revisar el apartado correspondiente dentro del tema General del Google Classroom del módulo.

Una vez se vaya a realizar la entrega, se comprimirá la carpeta del proyecto completo en un .rar o .zip, dejando el mismo nombre del proyecto seguido de la extensión del formato comprimido, por ejemplo **DII\_P2.1\_NOMBRE\_APELLIDO1\_APELLIDO2.rar**, y se subirá a la plataforma en la tarea de entrega correspondiente.

Solo se deberán implementar aquellos requisitos descritos en el enunciado. Antes de comprimir el proyecto para su entrega, se incluirá en la carpeta principal un fichero **README.txt** indicando que puntos se han desarrollado y cualquier otra explicación o comentario relevante que se considere, su objetivo no es para volver a escribir el enunciado.

Cualquier entrega que no cumpla cualquiera de los requisitos descritos anteriormente o que no esté entregada en tiempo y forma, no será corregida ni tenida en cuenta y tendrá que volverse a entregar en el período de entrega final de la evaluación.

No se resolverán dudas relativas a la presente práctica con menos de 24 horas de antelación a la fecha y hora de entrega de la misma.

En cualquier momento se podrá solicitar al alumno la defensa de la práctica entregada, ya sea a través de una exposición oral o mediante un vídeo demostrativo de su funcionamiento.

## 2. Enunciado

Modifica la Práctica 1.1 del Login para que haga uso de sesiones y siguiendo el patrón MVC.

Al iniciar la aplicación se mostrará el formulario de login (index.jsp) que contiene 2 cajas de texto (input): 1 de tipo texto (tipo text), para recoger el nombre del usuario (user), y 1 de tipo contraseña (tipo password), para recoger el valor de la contraseña (pass).

Así mismo se mostrará un enlace para que el usuario pueda darse de alta (alta.jsp), si no lo está ya, pudiendo logarse a continuación.

De la misma forma que se hacía en la Práctica 1.2 para los contactos, almacena los usuarios en una lista en memoria, por ejemplo un ArrayList de elementos Usuario.

Si el usuario intenta logarse y no está dado de alta previamente en el listado de usuarios (debes cargar datos de prueba al iniciar la aplicación), se redireccionará al formulario de alta (alta.jsp). El formulario de alta será el mismo empleado en la Práctica 1.1 con la doble validación de contraseña. Si el usuario intenta darse de alta con un nombre de usuario que ya existe, el sistema no realizará el alta y se informará de ello al usuario.

Si el usuario intenta logarse pero la contraseña no es correcta, debería informarse de ello al usuario en la misma pantalla del formulario de login.

Si el usuario se loga correctamente, se mantendrá su nombre a través de la sesión y se mostrará en la página de inicio junto a un enlace para poder cerrar la sesión.

Si el usuario cierra la sesión, se deberá volver a mostrar el formulario de login.

En la página de inicio del usuario (al logarse correctamente) se mostrará un enlace con la opción para poder cambiar la contraseña. Si el usuario pulsa dicho enlace, la aplicación conducirá a un formulario con 3 cajas de texto (input) de tipo contraseña: uno para recoger la contraseña anterior y otros dos para recoger el valor de la contraseña dos veces y realizar un doble check antes de realizar el cambio. Si la contraseña anterior es correcta y las dos contraseñas nuevas coinciden, se realizará la modificación y se

informará al usuario mediante un mensaje en la página de inicio del usuario, manteniéndose el nombre de usuario en la sesión en todo momento. Si la introducción de la contraseña no se realiza correctamente, se informará al usuario de la situación.

Si el usuario logado en el sistema es **admin** se mostrará un listado con los usuarios dados de alta almacenados en memoria. Para cada uno de los usuarios de la lista el usuario administrador podrá realizar las operaciones de borrado o edición de los datos del usuario (ÚNICAMENTE modificación del nombre, no de la contraseña). De la misma forma tendrá disponible el mismo enlace que un usuario sin permisos de administrador para modificar su propia contraseña.

Utiliza *includes* para proveer a la aplicación de una cabecera, donde se incluirá el css utilizado, y de un footer con el nombre del alumno. De esta forma todas las vistas tendrán la misma apariencia.

Incluye en el fichero README.txt los usuarios de prueba cargados y sus respectivas contraseñas para poder probar la aplicación.