

# Desarrollo de Interfaces

## Práctica 1.1

### 1. Formato y entrega

El proyecto se deberá desarrollar con el IDE Eclipse y tendrá el siguiente nombre: **DII\_P1.1\_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 ñes, 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.p11** (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\\_P1.1\\_NOMBRE\\_APELLIDO1\\_APELLIDO2](http://localhost:8080/DII_P1.1_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\_P1.1\_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

Desarrolla una aplicación web en Java que conste de un formulario para realizar un login siguiendo la siguiente descripción:

1. Como vista principal de la aplicación existe un fichero `index.jsp`, que se encuentra en el raíz del proyecto y al que se accede directamente a través de la URL base. Este fichero de entrada está formado por un formulario que contiene 3 cajas de texto (input), 1 de tipo texto (text), para recoger el nombre del usuario (user), y 2 de tipo contraseña (password), para recoger el valor de la contraseña dos veces y realizar un doble check (pass y pass2). Además cuenta con un botón para el envío de los datos al Servlet [2 puntos].
2. Se crea la clase de mapeo `Usuario`, con los atributos necesarios para almacenar los datos (3 campos) y poder operar con ellos posteriormente a través de los objetos de tipo `Usuario` que se crean con los datos recogidos del formulario [1 punto].
3. El Servlet recoge los datos del formulario, a través de los parámetros (atributo 'name' de los inputs), y se mapean con la clase `Usuario`, que contiene los atributos de tipo cadena user, pass y pass2 [1 punto].
4. En el caso de coincidir los valores de los dos campos de tipo contraseña, se muestra una nueva vista con un mensaje, saludando al usuario por su nombre (por ejemplo: "Hola X", siendo X el valor correspondiente al parámetro user, recogido a través del formulario de la vista principal) [2 puntos].
5. Si las contraseñas no coinciden, se muestra una nueva vista con un mensaje, advirtiéndole al usuario del error e invitándole a volver a introducir correctamente las contraseñas [1 punto].
6. Se modifica el punto anterior para que el mensaje de error se genere en la misma vista dónde se encuentre el formulario (**`index.jsp`**) [1 punto].
7. Se usa CSS en el diseño de la web [1 punto].

8. Se implementa funcionalidad adicional a la solicitada y/o se usan librerías de terceros de forma correcta y conveniente, siguiendo la documentación de la misma y siendo su uso coherente en el contexto de la aplicación (detallar en el fichero **README.txt** que debe acompañar a la aplicación) [1 punto].