

Desarrollo de Interfaces

Práctica 1.2

1. Formato y entrega

El proyecto se deberá desarrollar con el IDE Eclipse y tendrá el siguiente nombre: **DII_P1.2_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.p12** (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_P1.2_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_P1.2_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 simule una agenda de contactos:

1. Al cargar la url base del proyecto se mostrará un formulario con los campos correspondientes para almacenar un contacto:
 - Nombre [input-texto]
 - Apellidos [input-texto]
 - Email [input-email]
 - Teléfono [input-numérico]
 - Comentarios [textarea]

Además cuenta con un botón para el envío de los datos al Servlet.

Los datos serán validados en el servidor, por lo que si algún campo se deja sin rellenar (vacío) no se creará el contacto y se informará de ello al usuario mediante un mensaje en la misma vista dónde se encuentra el formulario (**index.jsp**) [2 puntos].

2. Se crea la clase POJO Contacto, con los atributos necesarios para almacenar los datos con sus correspondientes métodos, para poder operar con ellos posteriormente [1 punto].
3. Los datos recogidos del formulario se mapean con la clase Contacto y se almacenan en una lista en memoria, por ejemplo un ArrayList de elementos Contacto [1 punto].
4. Se crea una función para comprobar que el email del contacto introducido a través del formulario no existe ya en el listado de contactos. De ser así, el contacto se almacenará en el listado, en caso contrario, no se almacenará, informando al usuario con un mensaje en la vista [2 puntos].
5. Una vez almacenados los datos del nuevo contacto, mostrar en una vista, el listado de contactos almacenados en la lista en memoria [1 punto].

6. Se modifica el programa para que el listado de contactos aparezca en la misma vista dónde se encuentra el formulario (**index.jsp**) [1 punto].
7. Usar CSS para mostrar al usuario un mensaje en rojo informando en la vista si se ha podido almacenar o no el contacto [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].