

Para la realización de estas actividades podría ser necesario utilizar clases, métodos u opciones no vistos en el capítulo, por lo que se recomienda consultar la documentación de Java SE 8 (<https://docs.oracle.com/javase/8/docs/api/> )

### Precisiones adicionales:

Se entregará un único proyecto en Netbeans, compilable y ejecutable con JDK 1.8, con un menú numerado para cada ejercicio de los que se solicitan a continuación.

El código Java debe estar estructurado en Clases-métodos y comentado.

**NO SE SOLICITARÁN LOS DATOS AL USUARIO CUANDO SE CREEN DATOS A ALMACENAR EN LA BD, A MENOS QUE ASÍ SE SOLICITE EN EL EJERCICIO.** Los datos serán introducidos a través de un método propio dentro del programa. Pero se deberán informar de este hecho al usuario, para que, tras realizar la inserción, se pueda ir a la BD y comprobar su estado.

1. Crea un programa en Java que, a partir de los contenidos de un fichero XML, como el del ejemplo, con los datos de varios clientes, muestre un nuevo documento de XML que sea el resultado de añadir los datos de un nuevo cliente al principio, antes de todos los clientes que puedan existir. Debe utilizarse un *parser* DOM y las interfaces de DOM para añadir los datos del nuevo cliente, incluyendo DNI, apellidos y CP. La salida debe ser la salida estándar (*System.out*) y en formato XML, y debe generarse utilizando la clase *Transformer*. El fichero se debe especificar mediante un argumento de línea de comandos. Si el fichero no es un fichero XML correcto, o si lo es pero no tiene un único elemento con nombres de clientes en el nivel más alto, se debe terminar la ejecución inmediatamente y volver al menú principal, mostrando mensajes de error apropiados.
2. Crea una nueva opción cambiando el programa anterior para que la salida se realice al mismo fichero, y reemplace sus contenidos. La salida se debe generar también con la clase *Transformer*. Antes de nada hay que hacer una copia del fichero en un fichero con el mismo nombre pero incluyendo una marca de tiempo y al final ".bak". Se aconseja generar la salida en un fichero temporal, borrar el fichero original y renombrar el fichero temporal con el nombre del fichero original. En los programas de ejemplo vistos durante la unidad de ficheros ya se usó esta técnica. Se recomienda consultarlos y copiar y adaptar su código de programa. Si, como es lógico, no funciona a la primera, se tiene una copia del fichero original que se puede renombrar para intentarlo de nuevo.

3. Crea un programa como el anterior, pero que no permita añadir el nuevo cliente si en el fichero ya existe un cliente con su DNI. En este caso, debe mostrarse un mensaje de error y volver al menú principal. Además el cliente debe añadirse al final, en lugar de al principio.
4. Estructura el programa con las opciones anteriores basándote en una clase llamada `clientesXML` para encapsular convenientemente todas las operaciones. El método `main()` debe crear una instancia de esa clase y utilizarla para todas las operaciones. Su constructor debe tomar como argumento un nombre de fichero y obtener el documento DOM en una variable de clase de tipo `Document`. El constructor no debe gestionar ninguna excepción, de manera que debe añadirse en su declaración una cláusula `throws` con todos los tipos de excepción que sea necesario incluir. La inserción de un nuevo cliente debe realizarse mediante un método público de clase `insertaCliente(String DNI, String apellidos, String CP)`, y debe insertarse al final, después de todos los clientes que puedan existir. Este método debe devolver un `Node` con el nuevo nodo creado, o `null` si por la razón que fuera no se crea el nodo. No debe reflejar los cambios en el fichero, solo en el `Document`. No debe gestionar las excepciones que se puedan producir al insertar los datos del nuevo cliente en el `Document`, de manera que hay que añadir la cláusula `throws` apropiada. Debe tener un método `grabar()` que permita reemplazar los contenidos del fichero con los contenidos del documento DOM. La generación del fichero (serialización del documento DOM) se debe realizar con la clase `Transformer`. Este método debe, antes de nada, hacer una copia del fichero en un fichero cuyo nombre esté basado en el nombre original del fichero y que incluya una marca de tiempo con la fecha y hora. El programa de prueba debe insertar un cliente con un DNI ya existente en el momento en que se intenta la inserción y dos clientes cuyos DNI no existan. Si el DNI ya existe, debe mostrarse un mensaje de error y seguir adelante. El método `main()` debe quedar más o menos así:

```
CientesXML cXML = new CientesXML(nombreFichero);  
cXML.insertaCliente(dni1_nuevo, nombre1, CP1);  
cXML.insertaCliente(dni2_repetido, nombre2);  
cXML.insertaCliente(dni3_nuevo, nombre3, CP3);  
cXML.grabar();
```