

UA02. PRIMER ACCESO A DATOS

1.2. Creación de documentos con DOM

Para la creación de un documento DOM y paso a un fichero XML, usaremos nodos de distintos tipos con algunos métodos de la interfaz **Document**.

Se usarán estos para crear los nodos y métodos de la interfaz **Node** para añadirlos al documento.

```
1 | Element createElement(String nombreEtiqueta) // Crea nuevo elemento  
2 |  
3 | Text createTextNode(String texto) // Crea nuevo nodo de tipo texto
```

Los atributos se añaden con métodos de la interfaz **Element**.

```
1 | void setAttribute(String name, String value) // Añade atributo a elemento
```

Los siguientes métodos de la interfaz **Node**, permiten añadir elementos a un documento DOM. Si ocurre algún error, lanzará una excepción de tipo **DOMException**.

```
1 | Node appendChild(Node nuevoHijo) // Añade nodo como último nodo  
2 | Node insertBefore(Node nuevoHijo, Node refHijo) // Añade nuevo hijo antes de refHijo
```

Serializar un documento DOM

La clase **Transformer** permite serializar documentos DOM, es decir, generar un documento de texto XML a partir de ellos.

```
1  import java.io.StringWriter;
2  import javax.xml.parsers.DocumentBuilder;
3  import javax.xml.parsers.DocumentBuilderFactory;
4  import javax.xml.transform.OutputKeys;
5  import javax.xml.transform.Transformer;
6  import javax.xml.transform.TransformerFactory;
7  import javax.xml.transform.dom.DOMSource;
8  import javax.xml.transform.stream.StreamResult;
9
10
11  import org.w3c.dom.DOMImplementation;
12  import org.w3c.dom.Document;
13  import org.w3c.dom.Element;
14
15  public class CrearDocumentoDOM {
16      public static void main(String[] argv) throws Exception {
17          DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
18          DocumentBuilder db = dbf.newDocumentBuilder();
19          DOMImplementation impl = db.getDOMImplementation();
20          Document doc = impl.createDocument(null, null, null);
21
22          Element e1base = doc.createElement("clientes");
23          doc.appendChild(e1base);
24
25          Element e1 = doc.createElement("cliente");
26          e1.setAttribute("DNI", "12345678Z");
27          Element e1Apel = doc.createElement("apellidos");
28          e1Apel.appendChild(doc.createTextNode("ROJAS"));
29          e1.appendChild(e1Apel);
30          Element e1Validez = doc.createElement("validez");
31          e1Validez.setAttribute("estado", "borrado");
```

```
31     e1Validez.setAttribute("timeStamp", "1528286082");
32     e1.appendChild(e1Validez);
33     e1base.appendChild(e1);
34
35     DOMSource domSource = new DOMSource(doc);
36     Transformer transformer = TransformerFactory.newInstance().r
37     //transformer.setOutputProperty(OutputKeys.OMIT_XML_DECLARA1
38     transformer.setOutputProperty(OutputKeys.METHOD, "xml");
39     transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
40     transformer.setOutputProperty("{http://xml.apache.org/xslt}i
41     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
42     StringWriter sw = new StringWriter();
43     StreamResult sr = new StreamResult(sw);
44     transformer.transform(domSource, sr);
45     System.out.println(sw.toString());
46 }
47 }
```

Obra publicada con ~~Licencia~~ Creative Commons Reconocimiento Compartir igual 4.0
<<http://creativecommons.org/licenses/by-sa/4.0/>>