

```

1  package org.adrian.xml;
2
3  import lombok.NonNull;
4  import org.w3c.dom.Document;
5  import org.w3c.dom.Element;
6  import org.w3c.dom.Node;
7  import org.w3c.dom.NodeList;
8  import org.xml.sax.SAXException;
9
10 import javax.xml.parsers.DocumentBuilder;
11 import javax.xml.parsers.DocumentBuilderFactory;
12 import javax.xml.parsers.ParserConfigurationException;
13 import javax.xml.transform.*;
14 import javax.xml.transform.dom.DOMSource;
15 import javax.xml.transform.stream.StreamResult;
16 import java.io.File;
17 import java.io.IOException;
18 import java.util.ArrayList;
19 import java.util.List;
20
21 public class DOM {
22
23     //1º VARIABLES
24     private final String uri;
25     private static DOM controller;
26     private Document data;
27
28     //2º SINGLETON
29     private DOM(String uri) { //COMENTAR
30         this.uri = uri;
31     }
32     //Obtener instancia del DOM
33     public static DOM getInstance(@NonNull String uri) {
34         if (controller == null)
35             controller = new DOM(uri);
36         return controller;
37     }
38
39     //3º CARGAR LOS DATOS Obtencion de la informacion de un archivo según la
uri
40     public void loadData() throws ParserConfigurationException, IOException,
SAXException {
41         //File xmlFile = new File(this.uri); //Si la uri a leer es un XML y no un RSS
42         DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
43         DocumentBuilder dBuilder;
44         // Cargamos el documento normalizado
45         dBuilder = dbFactory.newDocumentBuilder();
46         this.data = dBuilder.parse(uri); //this.data = dBuilder.parse(xmlFile);
47         this.data.getDocumentElement().normalize();
48     }
49

```

```

50  //4º MOSTRAR EL DOM POR CONSOLA
51  public void printXML() throws TransformerException {
52      Transformer transformer= this.preProcess();
53      DOMSource source = new DOMSource(this.data);
54      StreamResult console = new StreamResult(System.out);
55      transformer.transform(source, console);
56  }
57
58  //5º PASAR LA INFORMACIÓN A UNA CLASE
59  //Obtencion de los hijos (item) de un Nodo del DOM
60  private Noticia getNoticia(@NonNull Node node) {
61      Noticia noticias = new Noticia();
62      for(Node n = node.getFirstChild(); n != null; n = n.getNextSibling()){ //
        IMPORTANTE
63          if (n.getNodeType() == Node.ELEMENT_NODE) {
64              if (n.getNodeName().equals("title"))
65                  noticias.setTitulo(n.getTextContent());
66              if (n.getNodeName().equals("category"))
67                  noticias.setCategoria(n.getTextContent());
68              if (n.getNodeName().equals("description"))
69                  noticias.setDescripcion(n.getTextContent());
70              if (n.getNodeName().equals("dc:creator"))
71                  noticias.setAuthor(n.getTextContent());
72              if (n.getNodeName().equals("pubDate"))
73                  noticias.setFecha(n.getTextContent());
74              if (n.getNodeName().equals("link"))
75                  noticias.setEnlace(n.getTextContent());
76          }
77      }
78      return noticias;
79  }
80  //Obtener lista de noticias del DOM
81  public List<Noticia> getNoticias() {
82      // Recorremos y obtenemos todos los usuarios obteniendo todos los nodos
        user reiniciando la lista
83      List<Noticia> noticiaList = new ArrayList<Noticia>();
84      NodeList nodeList = this.data.getElementsByTagName("item");
85      for (int i = 0; i < nodeList.getLength(); i++) {
86          noticiaList.add(getNoticia(nodeList.item(i)));
87      }
88      return noticiaList;
89  }
90  //-----
        -----
91  //6º CREACIÓN DE UN NUEVO ARCHIVO XML CON LA INFORMACIÓN DE
        NOTICIAS GUARDADA EN LA CLASE
92  //Creación de un fichero nuevo, vacio don guardar la información
93  public void initData() throws ParserConfigurationException {
94      DocumentBuilderFactory dbFactory = DocumentBuilderFactory.newInstance();
95      DocumentBuilder dBuilder;
96      dBuilder = dbFactory.newDocumentBuilder();

```

```

97     this.data = dBuilder.newDocument();
98     // Creamos el elemento raíz
99     Element rootElement = this.data.createElement("Noticias"); //Elemento padre
    a crear
100    // Lo añadimos al documento
101    this.data.appendChild(rootElement);
102 }
103
104 //7º SE AÑADEN LAS NOTICIAS DE NUESTRA CLASE
105 //CAMBIAR NOMBRES
106 private Node createNewsElements(String name, String value){
107     Element node = this.data.createElement(name);
108     node.appendChild(this.data.createTextNode(value));
109     return node;
110 }
111 private Node createNewsElement(Noticia noticia){
112     Element element = this.data.createElement("item");
113     element.appendChild(createNewsElements("title", noticia.getTitulo()));
114     element.appendChild(createNewsElements("category", noticia.getCategoria
115 ());
116     element.appendChild(createNewsElements("description", noticia.
getDescripcion()));
117     element.appendChild(createNewsElements("creator", noticia.getAuthor()));
118     element.appendChild(createNewsElements("pubDate", noticia.getFecha()));
119     element.appendChild(createNewsElements("link", noticia.getEnlace()));
120     return element;
121 }
122 public void addNewsItem(Noticia noticia){
123     Element rootElement = (Element) this.data.getElementsByTagName("
Noticias").item(0);
124     rootElement.appendChild(createNewsElement(noticia));
125 }
126
127 //8º SE ESCRIBE LA INFORMACIÓN EN UN XML
128 //PreProcesado del XML para imprimir (parámetros)
129 private Transformer preProcess() throws TransformerConfigurationException {
130     TransformerFactory transformerFactory = TransformerFactory.newInstance();
131     Transformer transformer = transformerFactory.newTransformer();
132     transformer.setOutputProperty(OutputKeys.INDENT, "yes");
133     return transformer;
134 }
135 //Guarda el DOM en el fichero xml indicado
136 public void writeXMLFile(String uri) throws TransformerException {
137     Transformer transformer= this.preProcess();
138     DOMSource source = new DOMSource(this.data);
139     StreamResult file = new StreamResult(new File(uri));
140     transformer.transform(source, file);
141     System.out.println("Fichero XML generado con éxito");
142 }
143

```