# Ejemplo con DOM y SAX

Para poder trabajar con DOM en Java necesitamos las clases e interfaces que componen el paquete org.w3c.dom (contenido en el JDK) y el paquete javax.xml.parsers del API estándar de Java que proporciona un par de clases abstractas que toda implementación DOM para Java debe extender. Además como DOM no define ningún mecanismo para generar un fichero XML a partir de un árbol DOM usamos javax.xml.transform.

## 1) utilizando DOM

*/* Realiza las siguientes tareas, a partir de la clase persona:Realiza las siguientes tareas, a partir de la clase persona: a) Crea un fichero «FichPersona.dat», que almacene varios objetos persona. b) Tomando como base el fichero anterior, crea un documento XML usando DOM. c) Implementa una clase que permita leer el documento XML del apartado anterior. */*

```
public class Persona implements Serializable{
private String nombre;
private int edad;

public Persona(String nombre, int edad){
this.nombre = nombre;
this.edad = edad;
}

public Persona(){
this.nombre = null;
}

public void setNombre(String nom){
nombre = nom;
}

public void setEdad(int ed){
edad = ed;
}

public String getNombre(){
return nombre;
}

public int getEdad(){
return edad;
}
```

```
public class ejemploDOM{

/*   Crea un fichero "FichPersona.dat", que almacene varios objetos persona.  */



private static void creaFicheroDatosPersonas(String[] args) throws IOException{

ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(new
File(args[0]), false));

oos.writeObject(new Persona("Pedro", 53));
oos.writeObject(new Persona("Juan", 26));
oos.writeObject(new Persona("Maria", 32));
oos.writeObject(new Persona("Laura", 19));

oos.close();
}

// Tomando como base el fichero anterior, crea un documento XML usando DOM.
private static void creaFicheroXMLPersonas(String[] args) throws IOException
{
FileInputStream fis = new FileInputStream(new File(args[0]));
ObjectInputStream ois = new ObjectInputStream(fis);
Persona perso;

try{
// Crea la cabecera del fichero XML en memoria:

Document document =
DocumentBuilderFactory.newInstance().newDocumentBuilder().getDOMImplementation().cr
eateDocument(null, "personas", null);
document.setXmlVersion("1.0");

// Añade los objetos persona al XML:
while( fis.available() > 0 ){
perso = (Persona)ois.readObject();

Element raiz = document.createElement("persona");
document.getDocumentElement().appendChild(raiz);
```

```
crearElemento("nombre", perso.getNombre(), raiz, document);
crearElemento("edad", String.valueOf(perso.getEdad()), raiz, document);
}



// Vuelca el XML a un fichero XML:

TransformerFactory.newInstance().newTransformer().transform(new
DOMSource(document), new StreamResult(new java.io.File(args[1])));

}catch( Exception e ){
e.printStackTrace();
}

ois.close();
fis.close();
}

static void crearElemento(String datoEmple, String valor, Element raiz, Document
document){
Element elem = document.createElement(datoEmple);
Text text = document.createTextNode(valor);
raiz.appendChild(elem);
elem.appendChild(text);
}

// Implementa un MÉTODO que permita leer el documento XML del apartado anterior.
public static void imprimirFicheroXMLPersonas(String[] args){
try{
Document document =
DocumentBuilderFactory.newInstance().newDocumentBuilder().parse(new File(args[1]));
document.getDocumentElement().normalize();
System.out.println(document.getDocumentElement().getNodeName());
NodeList empleados = document.getElementsByTagName("persona");

for( int i = 0 ; i < empleados.getLength() ; i++ ){
Node emple = empleados.item(i);
if( emple.getNodeType() == Node.ELEMENT_NODE ){
Element elemento = (Element)emple;

System.out.print(" " + getNodo("nombre", elemento));
System.out.println(" \t" + getNodo("edad", elemento));
}
}

}catch( Exception e ){
System.err.println("Error: " + e);
```

```
}
}

private static String getNodo(String etiqueta, Element elem){
NodeList nodo = elem.getElementsByTagName(etiqueta).item(0).getChildNodes();
Node valorNodo = (Node)nodo.item(0);
return valorNodo.getNodeValue();
}

public static void main(String[] args) throws IOException{
String[] argumentos = { "personas.dat", "personas.xml" };
creaFicheroDatosPersonas(argumentos);
creaFicheroXMLPersonas(argumentos);
imprimirFicheroXMLPersonas(argumentos);
}

}
```

## 2) utilizando SAX

```
/* Ejemplo con SAX.

Utiliza SAX para visualizar el contenido del fichero creado en el ejercicio anterior.  */



public static void main(String[] args) throws SAXException, IOException {
XMLReader lector = XMLReaderFactory.createXMLReader();
GestionContenido gestor = new GestionContenido();
lector.setContentHandler(gestor);
InputSource ficheroXML = new InputSource("PELICULAS.xml");
lector.parse(ficheroXML);
}
} //Fin del main

//Clase para sobrescribir los eventos:
class GestionContenido extends DefaultHandler{
public GestionContenido(){
super();
}
@Override
public void startDocument(){
System.out.println("Comienzo del documento");
}
```

```
@Override
public void endDocument(){
System.out.println("Fin del documento");
}


@Override
public void startElement(String uri,String nombre,String nombreC, Attributes atts){
System.out.println("\tPrincipio del Elemento: "+nombre);
}
@Override
public void endElement(String uri,String nombre,String nombreC){
System.out.println("\tFin del Elemento: "+nombre);
}
@Override
public void characters(char[]ch,int inicio,int longitud)throws SAXException{
String car = new String(ch,inicio,longitud);

//quita los saltos de linea:
car = car.replaceAll("[\t\n]","");
System.out.println("\tCaracteres: "+car);
}
}
```