

Grupo:

Frida Abella Fernández

Irene Alonso Suárez

Jaime Aranda Congil

David Matías Mota

ACTIVIDAD 4. ACCESO A DATOS. MANEJO DE XML

REQUERIMIENTO 1

En la clase Principal, creamos un método main que se encargará de crear el DOM o documento mediante los objetos de tipo DocumentBuilderFactory, DocumentBuilder y Document y sus métodos. Establecemos que el elemento raíz del documento sea “concierto”

```
public class Principal {  
  
    public static void main(String[] args) {  
        DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();  
        DocumentBuilder analizador;  
        Document doc;  
  
        try {  
            analizador = factory.newDocumentBuilder();  
            // Creamos nuevo documento o DOM  
            doc = analizador.newDocument();  
  
            // Añadimos elemento raíz  
            Element concierto = doc.createElement("concierto");  
            doc.appendChild(concierto);  

```

A continuación llamamos a los métodos agregarDatos() y guardar(), que servirán para añadir el resto de elementos o nodos hijos del documento y guardar el documento en formato XML, respectivamente. Los tendremos definidos o declarados fuera del main, a continuación. Todo ello irá encerrado en un bloque try/catch para evitar las excepciones y que saquen, si saltan, un mensaje de error.

```
            // Añadimos todo los datos al elemento raíz concierto:  
            agregarDatos(concierto, doc);  
  
            // Guardamos en disco el nuevo documento XML.  
            guardar(doc);  
  
            System.out.println("El archivo se ha creado con éxito");  
        } catch (Exception e) {  
            System.out.println(e.getMessage());  
        }  
    }  
}
```

A continuación, en el método `agregarDatos()`, procedemos a crear todos los nodos hijo del elemento raíz “conciertos”, en este caso los nodos hijo “fecha”, “hora” y “participantes”. Primero se crea el objeto de tipo `Element`, luego se le añade un nodo-hijo de tipo texto donde se define el texto que contiene ese elemento, y a continuación se anexiona este elemento al elemento raíz `conciertos`.

```
// Agregamos el primer nodo de texto y su contenido, y lo hacemos añadimos como :
Element fecha = doc.createElement("fecha");
fecha.appendChild(doc.createTextNode("20-oct-2018"));
concierto.appendChild(fecha);

// Creamos el nodo de texto de "hora"
Element hora = doc.createElement("hora");
hora.appendChild(doc.createTextNode("21:30"));
concierto.appendChild(hora);

// Agregamos el nodo-elemento de participantes, que contendrá varios
Element participantes = doc.createElement("participantes");
concierto.appendChild(participantes);
```

A continuación creamos el primer elemento o nodo hijo “participante” del nodo padre “participantes”. Lo anexionamos como nodo hijo. Después creamos los nodos hijo de este mismo elemento y se los adjuntamos con el método `.appendChild` como hemos hecho previamente a lo largo de todo el árbol DOM. Estos nodos-hijo serán: “entrada” y “grupo” y tendrán a su vez, respectivamente, nodos-hijo de tipo texto donde se incluye la información.

Se repite el proceso para cada uno de los tres nodos “participante” que cuelgan de “participantes”.

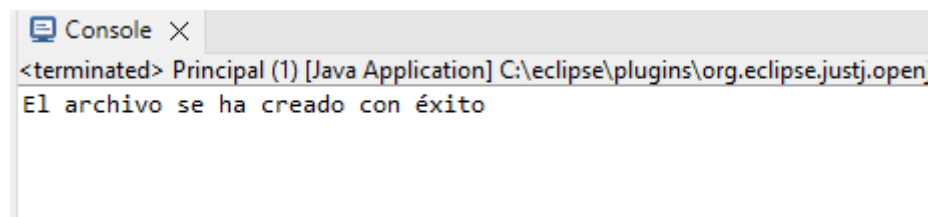
```
// Creamos el primer nodo-elemento participante
Element participante = doc.createElement("participante");
participantes.appendChild(participante);

// Creamos el nodo de texto de "entrada" dentro del primer participante
Element entrada = doc.createElement("entrada");
entrada.appendChild(doc.createTextNode("21:30"));
participante.appendChild(entrada);
// Creamos el nodo de texto de "grupo" dentro del primer participante
Element grupo = doc.createElement("grupo");
grupo.appendChild(doc.createTextNode("Las ardillas de Dakota"));
participante.appendChild(grupo);
```

Por último, definimos el método guardar(), que guardará en el documento “concierto.xml” un xml con los datos del DOM, mediante los objetos que sirven a este proceso llamados Transformer.

```
private static void guardar(Document doc) throws TransformerException {  
    // Creamos la fabrica de objetos para crear objeto Transformer  
    TransformerFactory fabricaConversor = TransformerFactory.newInstance();  
    // Creamos el objeto Transformer, que nos permitira serializar el arbol dom a un fichero  
    Transformer conversor = fabricaConversor.newTransformer();  
    // Creamos la fuente de la cual sacaremos el arbol dom  
    DOMSource fuente = new DOMSource(doc);  
    // Creamos el flujo de salida, al fichero que queremos  
    StreamResult resultado = new StreamResult(new File("concierto.xml"));  
    // Por ultimo, serializamos los datos  
    conversor.transform(fuente, resultado);  
}
```

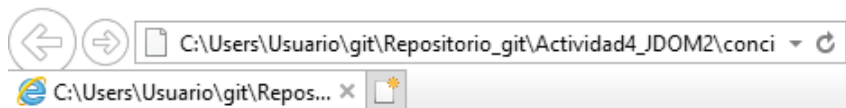
Al hacer Run As, nos saca por consola el mensaje de que el archivo se ha creado



Comprobamos que se ha creado, es el último de los ficheros que se observan aquí:

equipo > Disco local (C:) > Usuarios > Usuario > git > Repositorio_git > Actividad4_JDOM2				
Nombre	Fecha de modificación	Tipo	Tamaño	
.settings	03/03/2022 22:24	Carpeta de archivos		
bin	03/03/2022 22:24	Carpeta de archivos		
src	03/03/2022 22:24	Carpeta de archivos		
.classpath	03/03/2022 14:50	Archivo CLASSPATH	1 KB	
.gitignore	03/03/2022 14:52	Documento de te...	1 KB	
.project	03/03/2022 22:24	Archivo PROJECT	1 KB	
concierto	07/03/2022 0:23	Documento XML	1 KB	

Al abrirlo obtenemos el XML deseado:



```
<?xml version="1.0" encoding="UTF-8"?>
- <concierto>
  <fecha>20-oct-2018</fecha>
  <hora>21:30</hora>
  - <participantes>
    - <participante>
      <entrada>21:30</entrada>
      <grupo>Las ardillas de Dakota</grupo>
    </participante>
    - <participante>
      <entrada>22:15</entrada>
      <grupo>Fito y Fitipaldis</grupo>
    </participante>
    - <participante>
      <entrada>23:00</entrada>
      <grupo>Coldplay</grupo>
    </participante>
  </participantes>
</concierto>
```

Para la segunda parte del requerimiento 1, hemos creado otra clase, llamada LecturaXML.

Creamos el método main, donde a través de objetos de tipo DocumentBuilder vamos a poder parsear el contenido del documento xml que queremos leer.

Determinamos cuál será la raíz o elemento del DOM a partir del cuál se realice la lectura, mediante el método recorrerNodos(). Metemos todo ello dentro de un bloque try/catch para capturar excepciones.

```

public class LecturaXML {

    public static void main(String[] args) {
        DocumentBuilderFactory fabrica = DocumentBuilderFactory.newInstance();
        DocumentBuilder analizador;
        Document doc;
        Node raiz;

        try {
            analizador = fabrica.newDocumentBuilder();
            doc = analizador.parse("concierto.xml");
            raiz = doc.getDocumentElement();
            recorrerNodos(raiz);
        } catch (Exception e) {
            System.out.println(e.getMessage());
        }
    }
}

```

Con el método `recorrerNodos()` haremos una lectura de los métodos hijos que nacen directamente del elemento raíz, que como hemos definido en el main, es el elemento “conciertos”. Almacenamos todos en una lista u objeto de tipo `NodeList`.

A continuación, guardamos en objetos de tipo `Node` cada uno de los elementos, almacenados en las respectivas posiciones de la lista de nodos (0, 1 y 2).

Sacamos por pantalla su contenido a través del método `.getTextContent()` que accede al nodo-texto hijo que contienen.

Por último, llamamos dentro del método al otro método, `recorrerParticipantes()`.

```

private static void recorrerNodos(Node raiz) {
    //este metodo nos devuelve todos los nodos hijos Directos del elemento raiz "concierto"
    NodeList nodos = raiz.getChildNodes();

    // Dentro de esa lista de nodos, accedemos a los que se corresponden con los dos primeros, fecha y hora:
    Node fecha = nodos.item(0);
    Node hora = nodos.item(1);
    Node participantes = nodos.item(2);
    // lo sacamos por pantalla
    System.out.println("Fecha y hora del concierto: " + fecha.getTextContent() + " " + hora.getTextContent());
    recorrerParticipantes(participantes);
}

```

Desde el método `recorrerParticipantes()` iremos accediendo a los nodos hijo del elemento participantes, que son los 3 elementos “participante”. Al principio los almacenamos todos en una lista de nodos (`NodeList`) accediendo a ellos mediante el método `.getChildNodes`

A continuación recorreremos esa lista de nodos con un método `for`, guardando en un objeto `Node` el elemento afectado en ese momento, que deberá ser un “participante”

```
// Con el siguiente método hacemos una iteración para ir recorriendo uno a uno
private static void recorrerParticipantes (Node participantes) {
    // En esta lista obtenemos el conjunto de los elementos "participante"
    NodeList nodos = participantes.getChildNodes();
    System.out.println("Participan los siguientes grupos: ");
    for (int i=0; i<nodos.getLength();i++) {
        // En este objeto almacenamos el contenido del "participante" que es
        Node participante = nodos.item(i);
```

A continuación comprobamos que ese elemento afectado sea un nodo de tipo elemento (ya que no contiene texto, si no otros elementos dentro de sí). De ser true, accede a cada uno de los elementos dentro de participante y los guarda en un Node. Serán los elementos “entrada” y “grupo”

Imprimimos por pantalla accediendo a su contenido texto mediante el método .getTextContent()

```
if (participante.getNodeType() == Node.ELEMENT_NODE) {
    //dame los atributos del nodo escala, y quiero el atributo 0, es decir
    //el primer atributo, a continuación de pido el valor de ese nodo atributo
    Node entrada = participante.getChildNodes().item(0);
    Node grupo = participante.getChildNodes().item(1);
    System.out.print(" " + entrada.getTextContent() + "\n");
    System.out.print(" " + grupo.getTextContent() + "\n");
}
}
System.out.println();
```

Al ejecutar la aplicación, nos saca lo siguiente por consola:

```
Console X
<terminated> LecturaXML (1) [Java Application] C:\eclipse\plugins\
Fecha y hora del concierto: 20-oct-2018 21:30
Participan los siguientes grupos:
21:30
Las ardillas de Dakota
22:15
Fito y Fitipaldis
23:00
Coldplay
```

REQUERIMIENTO 2:

El objetivo ahora es trabajar con un XML pero con las librerías de JAXB.

En primer lugar creamos la clase Participante

```
package beans;

import javax.xml.bind.annotation.XmlElement;

@XmlRootElement(name="Participante")

@XmlType(propOrder = {
    "entrada",
    "grupo",
})

public class Participante {
```

Con la anotación `@XmlRootElement` le damos el nombre del nodo raíz y es obligatoria.

Con la anotación `@XmlType` establecemos el orden de como queremos que salga nuestro XML.

Creamos la clase Concierto donde haremos un `arrayList` de Participantes con lo que esta tendría una relación de 1-N ya que podría tener tantos participantes como se desee y concierto solo habrá uno.

```

package beans;

import java.util.ArrayList;

@XmlRootElement(name="Concierto")

public class Concierto {

    private String fecha;
    private String hora;
    private List <Participante> participante;

```

Creamos la clase ObjectToXML para a partir de los objetos crear un XML.

```

    */
    m.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);

    Concierto concierto = new Concierto();
    concierto.setFecha("22-OCT-2022");
    concierto.setHora("22:00");
    concierto.getParticipante().add(new Participante("22:30", "Green Day"));
    concierto.getParticipante().add(new Participante("23:15", "The Offspring"));
    concierto.getParticipante().add(new Participante("00:00", "Simple Plan"));
    concierto.getParticipante().add(new Participante("00:45", "Coldplay"));
    concierto.getParticipante().add(new Participante("01:15", "U2"));

    //Convertimos un objeto a xml y lo imprimimos por pantalla
    m.marshal(concierto, System.out);
    //tambien podemos crear un fichero
    m.marshal(concierto, new File("conciertos.xml"));
    System.out.println("Se ha creado con éxito el fichero");
} catch (JAXBException e) {
    System.out.println("Error convirtiendo el objeto a formato XML");
    System.out.println(e.getMessage());
    e.printStackTrace();
}

```


Lo imprimiremos por pantalla:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Concierto>
  <fecha>22-OCT-2022</fecha>
  <hora>22:00</hora>
  <participante>
    <entrada>22:30</entrada>
    <grupo>Green Day</grupo>
  </participante>
  <participante>
    <entrada>23:15</entrada>
    <grupo>The Offspring</grupo>
  </participante>
  <participante>
    <entrada>00:00</entrada>
    <grupo>Simple Plan</grupo>
  </participante>
  <participante>
    <entrada>00:45</entrada>
    <grupo>Coldplay</grupo>
  </participante>
  <participante>
    <entrada>01:15</entrada>
    <grupo>U2</grupo>
  </participante>
</Concierto>
Se ha creado con éxito el fichero
```

Y a continuación mostramos el XML creado:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<Concierto>
  <fecha>22-OCT-2022</fecha>
  <hora>22:00</hora>
  <participante>
    <entrada>22:30</entrada>
    <grupo>Green Day</grupo>
  </participante>
  <participante>
    <entrada>23:15</entrada>
    <grupo>The Offspring</grupo>
  </participante>
  <participante>
    <entrada>00:00</entrada>
    <grupo>Simple Plan</grupo>
  </participante>
  <participante>
    <entrada>00:45</entrada>
    <grupo>Coldplay</grupo>
  </participante>
  <participante>
    <entrada>01:15</entrada>
    <grupo>U2</grupo>
  </participante>
</Concierto>
```

Por último creamos la clase XMLToObject para así a partir del XML creado obtener un objeto:

```
public class XMLToObject {

    public static void main(String[] args) {

        try {
            JAXBContext contexto = JAXBContext.newInstance(Concierto.class);

            //Esta vez creamos un objeto que nos permite pasar
            //de XML a Object, es decir deserializar

            Unmarshaller u = contexto.createUnmarshaller();
            File fichero = new File("conciertos.xml");
            if (fichero.exists()) {
                Concierto c = (Concierto)u.unmarshal(fichero);

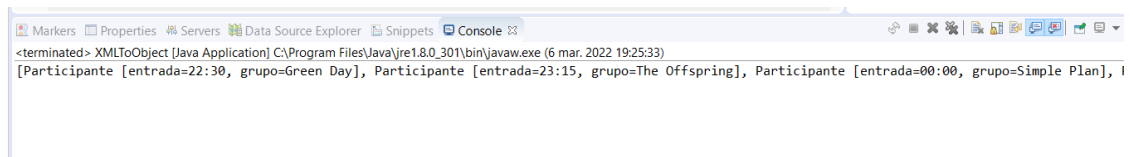
                System.out.println(c.getParticipante());
            } else {
                System.out.println("Fichero XML Concierto.xml no encontrado");
            }

        } catch (JAXBException e) {
            System.out.println(e.getMessage());
        }

    }

}
```

A continuación podemos ver que lo ha encontrado y lo muestra por pantalla:



```
<terminated> XMLToObject [Java Application] C:\Program Files\Java\jre1.8.0_301\bin\javaw.exe (6 mar. 2022 19:25:33)
[Participante [entrada=22:30, grupo=Green Day], Participante [entrada=23:15, grupo=The Offspring], Participante [entrada=00:00, grupo=Simple Plan], I
```