

ReadMe

Contenido

Requerimiento 1	2
1.GeneracionXML.java	2
Salida por pantalla.....	2
2.LecturaXML.java	2
Salida por pantalla.....	3
Requerimiento 2	3
1. Application.....	3
1. Application	3
2.Application.modelo.entidad.....	4
1. Catálogo.....	4
2. Autor	4
3. Editorial	4
4. Librería	4
5. Libro	4
3.Application.modelo.persistencia	4
Salida por pantalla.....	4
Ejecutables	6
GitHub	6

Requerimiento 1

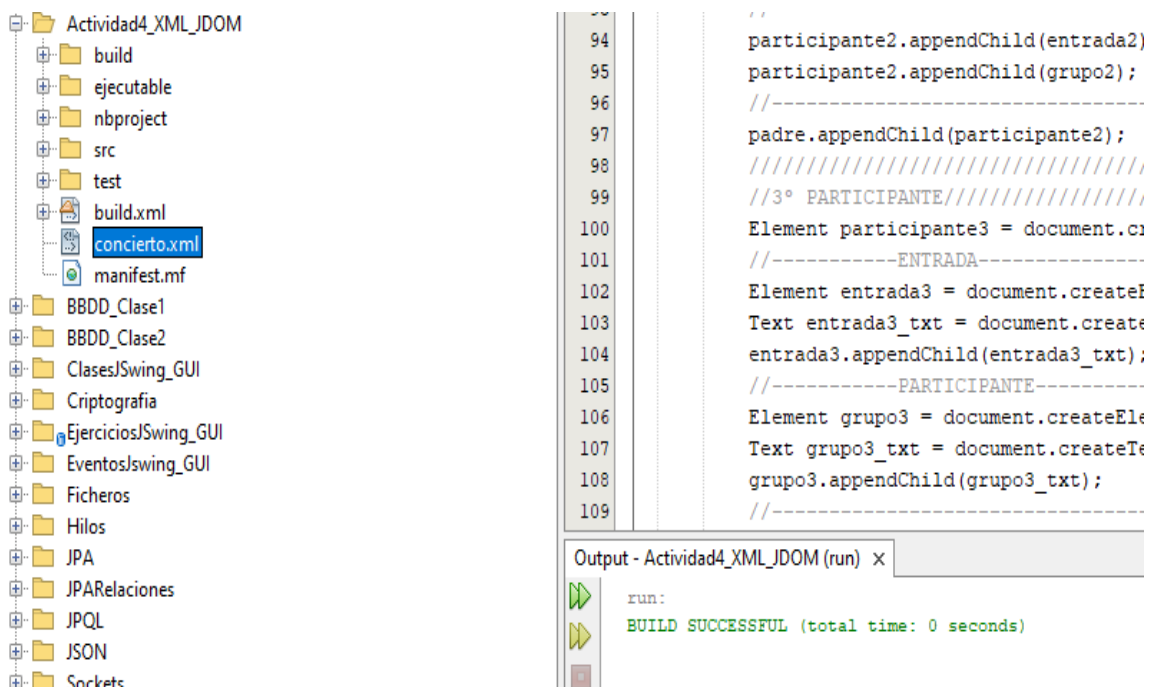
El requerimiento 1 se ha llevado a cabo mediante la librería JDOM; y consta de las siguientes clases:

1. GeneracionXML.java

Esta clase consta de los siguientes métodos:

1. **public static void main(String[] args):** Este es el método principal; y se encarga de generar un fichero xml llamado “concierto.xml” en la misma carpeta donde el programa en cuestión sea ejecutado.
2. **agregarElementosHijos(Element padre, Document document):** Este método se encarga de asignar nodos hijos, con su respectivo texto, a un nodo padre pasado como parámetro.
3. **generarXML(Document doc):** Este método, se encarga de generar el documento xml final llamado “concierto.xml”.

Salida por pantalla:



2. LecturaXML.java

Esta clase consta de los siguientes métodos:

4. **public static void main(String[] args):** Este es el método principal; y se encarga de leer y mostrar por pantalla un fichero xml llamado “concierto.xml” ,

que deberá estar alojado en el mismo directorio donde el programa en cuestión sea ejecutado.

5. **pintarFechaHora(Node raiz):** Este método devuelve una cadena de texto con la fecha y la hora en la que se celebrará el concierto del fichero “concierto.xml”. Recibe como parámetro el nodo raíz de dicho documento, el cual es recorrido hasta encontrar los nodos hijos necesarios.
6. **pintarParticipantes(Node raiz):** Este método devuelve una cadena de texto con todos los participantes del concierto del fichero “concierto.xml”; y la hora a la que tocarán. Recibe como parámetro el nodo raíz de dicho documento, el cual es recorrido hasta encontrar los nodos hijos necesarios.
7. **guardarFichetoTxt(String string):** Este método recibe como parámetro una variable de tipo String, donde se recogerá toda la información del concierto, para ser persistida en otro fichero llamado “concierto.txt”.

Salida por pantalla:

```
96
97
98 //Obtenemos a los participantes
99 for (int i = 0; i < participantes.getLength();
100 //
101 if (participantes.item(i).getNodeName().equ
102 participante = participantes.item(i).ge
103 for (int j = 0; j < participante.getLen
104 if (participante.item(j).getNodeTyp
105 if (participante.item(j).getNod
106 p = p + "-" + participante
107 }
108 if (participante.item(j).getNod
109 p = p + participante.item(j)
110 }
```

Output - Actividad4_XML_JDOM (run) x

```
run:
Fecha y hora del concierto: 20-Oct-2018 a las 21:30.
Participan los siguientes grupos:
- 21:30 Las Ardillas de Dakota.
- 22:15 Fito y Fitipaldis.
- 23:00 Coldplay.

BUILD SUCCESSFUL (total time: 0 seconds)
```

Requerimiento 2

El requerimiento 2 se ha llevado a cabo mediante la librería JAXB; y consta de los siguientes paquetes:

1. Application

Este paquete consta de las siguientes clases:

1. Application: Esta clase contiene el método main; y se encarga de generar un fichero xml llamado “catalogo.xml” en la misma carpeta donde el programa en cuestión sea ejecutado.

2.Application.modelo.entidad

Este paquete aloja las clases que constituirán cada uno de los nodos del fichero “catalogo.xml”, con sus respectivas relaciones. Dicho fichero consiste en un catálogo de libros, clasificados por autor, editorial y librería en la que se encuentran disponibles.

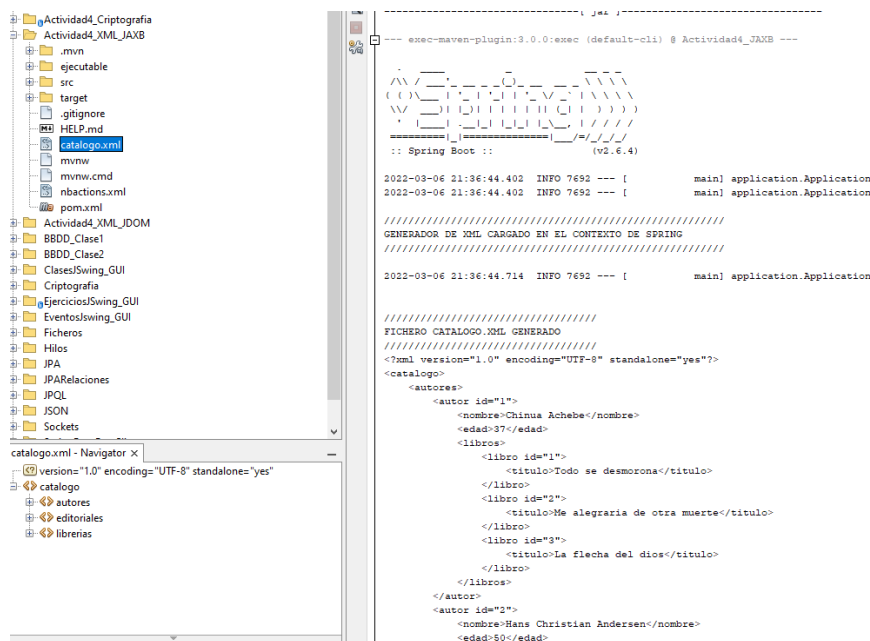
1. Catálogo: Nodo raíz.
2. Autor – Libros: Relación 1 a N.
3. Editorial – Libros: Relación 1 a N.
4. Librería – Libros: Relación 1 a N.
5. Libro: Nodo hijo.

3.Application.modelo.persistencia

Este paquete consta de las siguientes clases:

1. **Controlador_XML:** Esta clase contiene todos los métodos necesarios para serializar y deserializar un objeto en formato xml.

Salida por pantalla:



Actividad4_XML_JAXB

- .mvn
- ejecutable
- src
- target
- .gitignore
- HELP.md
- catalogo.xml**
- mvnw
- mvnw.cmd
- nactions.xml
- pom.xml

- Actividad4_XML_JDOM
- BBDD_Clas1
- BBDD_Clas2
- ClasesJswing_GUI
- Criptografia
- EjerciciosJswing_GUI
- EventosJswing_GUI
- Ficheros
- Hilos
- JPA
- JPARelaciones
- JPQL
- JSON
- Sockets

catalogo.xml - Navigator ×

```
.....<?xml version="1.0" encoding="UTF-8" standalone="yes"
```

```
LIBRO{id=2, titulo=Me alegria de oír muerte;
Libro{id=3, titulo=La flecha del dios}
//////////
```

```

////EDITORIAL ---> Fandogamia Editorial
Libro{id=4, titulo=Termiteros de la sabana}
Libro{id=5, titulo=La capsa de contes}
Libro{id=6, titulo=El patito feo}
//////////

```

```

////EDITORIAL ---> Planeta Comic
Libro{id=7, titulo=Pulgarcita}
Libro{id=8, titulo=Cuentos de Andersen}
Libro{id=9, titulo=Microlitos}
//////////

```

```

////LIBRERIA ---> 2Pajaros
Libro{id=1, titulo=Todo se desmorona}
Libro{id=2, titulo=Me alegraria de otra muerte}
Libro{id=3, titulo=La flecha del dios}
////

```

```

////LIBRERIA ---> Abarzuza
Libro{id=4, titulo=Termiteros de la sabana}
Libro{id=5, titulo=La capsa de contes}
Libro{id=6, titulo=El patito feo}
////

```

```

LIBRERIA ---> Acuarel xxx
Libro{id=7, titulo=Pulgarcita}
Libro{id=8, titulo=Cuentos de Andersen}
Libro{id=9, titulo=Microlitos}
//////////

```

//////////////////FIN DEL PROGRAMA////////////////////////

BUILD SUCCESS

Total time: 1.515 s

Ejecutables

Cada requerimiento conta de su respectivo ejecutable, alojado en los siguientes directorios:

1. Actividad4_XML\Requerimiento1\Actividad4_XML_JDOM\ejecutable\ejecutable.jar
2. D:\PROYECTOS_JAVA\Actividad4_XML\Requerimiento2\Actividad4_XML_JAXB\ejecutable\ ejecutable.jar

GitHub: https://github.com/GonzaloSC95/Actividad4_XML.git