

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

ENUNCIADO 18: Leer XML

Escribe el código necesario para poder leer un documento XML. En este caso será el fichero clase.xml que lo situaremos en el paquete resources y que contiene la siguiente estructura

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<clase>
  <alumno numero = "393">
    <nombre> Luis </nombre>
    <apellido> Luna </apellido>
    <apodo> Na </apodo>
    <marcas> 85 </marcas>
  </alumno>
  <alumno numero = "493">
    <nombre> Antonio </nombre>
    <apellido> Alvarez </apellido>
    <apodo> Avez </apodo>
    <marcas> 95 </marcas>
  </alumno>
  <alumno numero = "593">
    <nombre> Juan </nombre>
    <apellido> Juz </apellido>
    <apodo> jazz </apodo>
    <marcas> 90 </marcas>
  </alumno>
</clase>
```

ENUNCIADO 19: Leer XML

En este ejercicio se visualizar el mismo fichero que el enunciado 18 pero mostrando los datos por empleado en horizontal. Nota del profesor:” *Preferiblemente prefiero este criterio de visualización en horizontal. Que no te importe si no está bien tabulado. Es tema de la consola*”

ENUNCIADO 20: Leer XML con elementos con un número de atributos variable y elementos opcionales.

Escribe el código necesario para poder leer un documento XML . En este caso será el fichero clase2.xml que lo situaremos en el paquete resources y que contiene la siguiente estructura:

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<clase>
  <alumno numero = "393" dni="11111111A">
    <nombre> Luis </nombre>
    <apellido> Luna </apellido>
    <apodo> Na </apodo>
    <marcas> 85 </marcas>
  </alumno>
  <alumno numero = "493">
    <nombre> Antonio </nombre>
    <apellido> Alvarez </apellido>
    <apodo familiar="campos"> Avez </apodo>
    <marcas> 95 </marcas>
  </alumno>
  <alumno numero = "593">
    <nombre> Juan </nombre>
    <apellido> Juz </apellido>
    <apodo> jazz </apodo>
    <marcas> 90 </marcas>
    <delegado/>
  </alumno>
</clase>
```

1

2

3

En el punto 1: observa como no todos los alumnos tienen el mismo número de atributos.

En el punto 2: observa que el la etiqueta apodo puede tener atributos.

En el punto 3: observa que hay etiquetas que no están en todos los alumnos.

ENUNCIADO 21: Escribir XML

Escribe un programa que permita escribir un fichero XML (Escritoras.xml) utilizando DOM. El fichero tendrá este formato.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?><Escritoras>
  <Escritora>
    <nombre fechanac="1861">Sofia casanova</nombre>
  </Escritora>
  <Escritora>
    <nombre fechanac="1837">Rosalia de castro</nombre>
  </Escritora>
</Escritoras>
```

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

ENUNCIADO 22: Escribir xml con incorporación de Métodos.

Se propone escribir un fichero xml con el siguiente formato:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Escritoras>
  <Escritora>
    <nombre fechanac="1861">Sofia casanova</nombre>
    <lugarnacimiento>Almeiras</lugarnacimiento>
  </Escritora>
  <Escritora>
    <nombre fechanac="1837">Rosalia de castro</nombre>
    <lugarnacimiento>Padron</lugarnacimiento>
  </Escritora>
</Escritoras>
```

pero para ello deberás utilizar el método que se indica para simplificar el proceso:

```
static void crearElemento(String dato, String valor, Element raiz, Document document, ArrayList<Atributo> listaAtrib) {
    Element elem = document.createElement(tagName: dato);
    elem.setTextContent(textContent: valor);
    raiz.appendChild(newChild: elem);
    for (Atributo atributo : listaAtrib) {
        elem.setAttribute(name: atributo.getDato(), value: atributo.getValor());
    }
}
```

El primer parámetro es el nombre de la etiqueta a crear, el segundo el valor que tendría (null si es vacío), el tercer parámetro cual es su elemento padre, el cuarto el documento de trabajo y por último una lista de los atributos que podría llevar el elemento.

Ejemplo de código.

```
ArrayList<Atributo> listaatributos = new ArrayList<>();
NodeList nodosescritoras = raiz.getElementsByTagName(name: "Escritora");

crearElemento(dato: "Escritora", valor: null, raiz, document, listaAtrib: listaatributos);
listaatributos.add(new Atributo(d: "fechanac", v: "1861"));
crearElemento(dato: "nombre", valor: "Sofia casanova", (Element) nodosescritoras.item(index: 0), document, listaAtrib: listaatributos);
crearElemento(dato: "lugarnacimiento", valor: "Almeiras", (Element) nodosescritoras.item(index: 0), document, new ArrayList<>());
```

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

ENUNCIADO 23: Escribir XML final

Se propone escribir un fichero xml con el siguiente formato:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Personajes>
  <Personaje fechanac="1861" id="1">
    <nombre>Sofia casanova</nombre>
    <profesion>Novelista</profesion>
    <lugarnacimiento>Almeiras</lugarnacimiento>
  </Personaje>
  <Personaje fechanac="1837" id="2">
    <nombre>Rosalia de castro</nombre>
    <profesion>Escritora</profesion>
    <lugarnacimiento/>
  </Personaje>
  <Personaje fechanac="1958" id="3">
    <nombre>Maria José Alonso</nombre>
    <profesion>Científica</profesion>
    <lugarnacimiento>Leon</lugarnacimiento>
    <viva/>
  </Personaje>
</Personajes>
```

Podrás utilizar el método propuesto de crear elemento o no , pero por comodidad se recomienda.

ENUNCIADO 24: Leer SAX

Escribe un programa que permita leer el fichero clase.xml y obtener esta salida.

```
-----
Comienzo del documento XML
Numero :393      nombre: Luis      apellido: Luna      apodo: Na      marcas: 85
Numero :493      nombre: Antonio     apellido: Alvarez   apodo: Avez      marcas: 95
Numero :593      nombre: Juan      apellido: Juz      apodo: jazz      marcas: 90
Total Alumnos: 3
```

ENUNCIADO 25: Leer STAX. API CURSOR

Escribe un programa que permita leer el fichero clase.xml y obtener esta salida.

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

```
numero393    nombre: Luis    apellido: Luna    apodo: Na    marcas: 85
numero493    nombre: Antonio    apellido: Alvarez    apodo: Avez    marcas: 95
numero593    nombre: Juan    apellido: Juz    apodo: jazz    marcas: 90
Total Alumnos 3
```

Nota: no se ha echo un ejemplo con API event porque la idea de leer un xml es la misma.

ENUNCIADO 26: Crear un XML. JAXB. PROCESO MARSHALL

Escribe un programa que permita crear un fichero empleadosjaxb.xml a partir del empleadosjaxb.xsd que se te proporciona. El fichero tendrá esta estructura.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<empleados>
  <empleado>
    <id>100</id>
    <nombre>Patricia</nombre>
    <dep>88</dep>
    <salario>1000.1</salario>
  </empleado>
  <empleado>
    <id>200</id>
    <nombre>Lucia</nombre>
    <dep>99</dep>
    <salario>1000.2</salario>
  </empleado>
</empleados>
```

```
JAXBContext contexto = JAXBContext.newInstance(classesToBeBound: Empleados.class);
Marshaller marshallerObj = contexto.createMarshaller();
marshallerObj.marshal(o: listaEmpleados, new File(pathname: "./src/main/resources/nuevosempleadosjaxb.xml"));
```

Recuerda que previamente debes realizar el proceso que genera automáticamente el modelo (POJOS) a partir del xsd. Para serializar lo que harás será crear una lista en memoria y posteriormente con el marshall generar el fichero.

ENUNCIADO 27: Leer un XML. JAXB. PROCESO UNMARSHALL

Lee el fichero previamente creado. Para ello realizarás el proceso UNMARSHALL que pasa el fichero a memoria y lo lee.

ACCESO A DATOS DISTANCIA

UD1: MANEJO DE FICHEROS

```
JAXBContext contexto=JAXBContext.newInstance(classesToBeBound: Empleados.class);  
Unmarshaller u=contexto.createUnmarshaller();  
Empleados listaEmpleados = (Empleados) u.unmarshal(new File (pathname: "./src/main/resources/empleadosjaxb.xml"));
```

ENUNCIADO 28: Operaciones con JAXB. Insertar, Borrar, Modificar

Realiza un programa que a través de un menú permita operar con el fichero empleadosjaxb.xml y permita insertar un empleado, borrar un empleado y modificar un empleado.