I. Ficheros de intercambio

Intercambio de información

A principios de los años 90 surgió el problema de que los ordenadores pudieran entenderse entre sí. Estos ordenadores utilizaban diferentes sistemas operativos y sus programas estaban escritos en diferentes lenguajes de programación.

Existen diversos formatos para poder intercambiar información entre sistemas, independientemente del sistema operativo o del lenguaje de programación con el que estuviera hecho el software.

- CSV
- XML
- JSON

CSV

Los archivos CSV (Comma-separated values) son ideales para el intercambio de grandes cantidades de datos, ya que son más fáciles de transmitir y ocupan mucho menos espacio que los ficheros XML o JSON.

```
id,nombre,ciclo,curso,profesor
ad,Acceso a Datos,DAM,2,Alejandro
psp,Programación de servicios y procesos,DAM,2,David
pro,Programación,DAW,1,Ana
```

XML

Los archivos XML (eXtensible Markup Language) usan etiquetas para darle significado a los valores que almacenan. Forman una estructura de árbol donde los elementos pueden tener contenido textual y atributos.

```
<modulos>
    <modulo id="ad">
        <nombre>Acceso a Datos</nombre>
        <ciclo>DAM</ciclo>
        <curso>2</curso>
        <profesor>Alejandro</profesor>
    </modulo>
    ...
</modulos>
```

JSON

Los archivos JSON (JavaScript Object Notation) se representan como pares de clave y valor en un formato jerárquico semiestructurado. A diferencia de los XML, los JSON tienden a ser más pequeños, siendo populares en el intercambio de datos en línea.

{“modulos”:[{“id”:“ad”,“nombre”:“Acceso a Datos”,
“ciclo”:“DAM”, “curso”:2,“profesor”:“Alejandro”}

{“id”:“psp”,“nombre”:“Programación de servicios y procesos”,
“ciclo”:“DAM”, “curso”:2,“profesor”:“David”}

{“id”:“pro”,“nombre”:“Programación”,
“ciclo”:“DAW”, “curso”:1,“profesor”:“Ana”}]}

CSV

Lectura de ficheros CSV

Files.lines en java.nio

```
List<List<String>> libros = Files.lines(Paths.get("libros.csv"))
        .map(linea -> Arrays.asList(linea.split(COMMA_DELIMITER)))
        .toList();
```

Lectura de ficheros CSV

BufferedReader en java.io

```
List<List<String>> libros = new ArrayList<>();

try (FileReader fr = new FileReader("libros.csv");
      BufferedReader br = new BufferedReader(fr)) {
       String linea;
       while ((linea = br.readLine()) != null) {
             String[] libro = linea.split(COMMA_DELIMITER);
             libros.add(Arrays.asList(libro));
       }
} catch (IOException e) {
      e.printStackTrace();
}
```

Lectura de ficheros CSV

Librería OpenCSV

```
List<List<String>> libros = new ArrayList<>();

try (FileReader fr = new FileReader("libros.csv");
      CSVReader csvReader = new CSVReader(fr)) {
       String[] libro;
       while ((libro = csvReader.readNext()) != null) {
             libros.add(Arrays.asList(libro));
       }
} catch (IOException e) {
      e.printStackTrace();
}
```

XML

XML

Los archivos XML (eXtensible Markup Language) permiten el intercambio de información entre aplicaciones usando un fichero de texto plano al que se le pueden añadir etiquetas para darle significado a los valores que almacenan.

Un fichero XML se divide en dos partes:

- prolog: metadatos administrativos (declaración XML, procesamiento opciona tipo de documento...)
- body: se compone de dos partes: estructural y de contenido

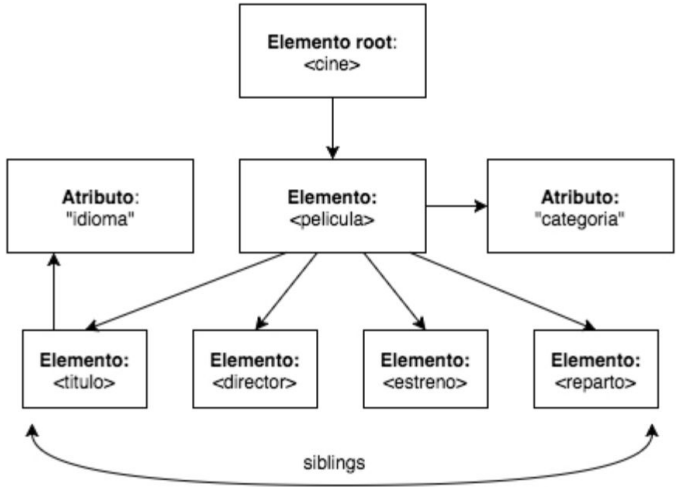
```xml
<?xml version="1.0" encoding="UTF-8"?>
<cine>
  <pelicula categoria="accion">
    <titulo idioma="ingles">Mad Max</titulo>
    <director>George Miller</director>
    <estreno>15 mayo 2015</estreno>
    <reparto>Tom Hardy</reparto>
    <reparto>Charlize Theron</reparto>
    <reparto>Nicholas Hoult</reparto>
  </pelicula>
  <pelicula categoria="animacion">
    <titulo idioma="ingles">Inside Out</titulo>
    <director>Pete Docter</director>
    <estreno>17 junio 2015</estreno>
    <reparto>Amy Poehler</reparto>
    <reparto>Phyllis Smith</reparto>
    <reparto>Bill Hader</reparto>
  </pelicula>
</cine>
```

XML

Los archivos XML forman una estructura de tipo árbol, comenzando desde la raíz (root), con ramas (branches) hacia las hojas (leaves).

Se emplean los términos parent, child y sibling para determinar parentesco: padre, hijo y hermano.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<cine>
  <pelicula categoria="accion">
    <titulo idioma="ingles">Mad Max</titulo>
    <director>George Miller</director>
    <estreno>15 mayo 2015</estreno>
    <reparto>Tom Hardy</reparto>
    <reparto>Charlize Theron</reparto>
    <reparto>Nicholas Hoult</reparto>
  </pelicula>
  <pelicula categoria="animacion">
    <titulo idioma="ingles">Inside Out</titulo>
    <director>Pete Docter</director>
    <estreno>17 junio 2015</estreno>
    <reparto>Amy Poehler</reparto>
    <reparto>Phyllis Smith</reparto>
    <reparto>Bill Hader</reparto>
  </pelicula>
</cine>
```



XML

Existen varias librerías para manipular ficheros XML desde Java:

- DOM (Document Object Model). Carga el árbol del fichero en memoria. Trabajamos nodo a nodo. Recomendable para ficheros pequeños.
- SAX (Simple API for XML). No carga el fichero en memoria, sino que recorre el fichero de forma secuencial informando de la ocurrencia de eventos, como el comienzo de un elemento XML o el final del fichero.
- JDOM: Nos permite leer, editar y escribir XML muy fácilmente.
- JAXB (Java Architecture for XML Binding). Define como los objetos son convertidos en Java desde/a XML.

JAXB

JAXB permite mapear clases Java a representaciones en XML y viceversa.

Proporciona dos características principales

- Serialización (marshalling) de objetos Java a XML.
- Deserialización (unmarshalling) de XML a objetos Java.

Es decir, JAXB permite almacenar y recuperar datos en memoria en cualquier formato XML, sin la necesidad de implementar un conjunto específico de rutinas XML de carga y salvaguarda para la estructura de clases del programa.

JSON

JSON

Los archivos JSON (JavaScript Object Notation) es un formato ligero para el intercambio de datos de forma estructurada.

Cada vez más popular en lugares donde antes se empleaba XML.

- Objeto: Un objeto JSON es una colección de pares nombre : valor
- Array: Un array JSON es una colección ordenada de valores separados por “,”

```
[
  {
    "nombre": "Pepito Conejo",
    "edad": 25,
    "carnet de conducir": true
  },
  {
    "nombre": "Ana Barberá",
    "edad": 90,
    "carnet de conducir": false
  }
]
```

JSON vs XML

JSON y XML son:

- Autodescriptivos (legibles por un humano)
- Jerárquico (valores dentro de valores)
- Parseables por muchos lenguajes

A diferencia de XML, JSON

- No usa etiquetas
- Es más corto
- Más rápido de leer y escribir
- No usa etiquetas

XML es más difícil de parsear que JSON!

JSON

Existen varias librerías para manipular ficheros JSON desde Java. Las principales son:

- Jackson
  - https://github.com/FasterXML/jackson
  - Es la biblioteca por defecto en Spring para JSON.
- GSon
  - https://github.com/google/gson
  - Biblioteca de JSON de Google. Es más ligera que Jackson.