

ACCESO A DATOS

UD02: PRÁCTICA 1 - XML: DOM

Proyecto: adt2_practica1

Requisitos previos:

- Tener instalada la librería estándar de Java (no hace falta ninguna extra).
- Comprender la estructura básica de un XML.
- Importar las clases del paquete javax.xml.parsers y org.w3c.dom.

Cada ejercicio debe estar en un paquete distinto con su clase AppX.java y método main().

Carpeta de recursos: ./resources

- ejercicio1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<libros>
  <prestamo>
    <libro isbn="9734602">
      <titulo>El Ocho</titulo>
      <autor>Katherine Neville</autor>
      <anyo>1988</anyo>
      <editorial>Ballantine Books</editorial>
    </libro>
    <libro isbn="98368520">
      <titulo>Qumrán</titulo>
      <autor>Eliette Abécasis</autor>
      <anyo>1997</anyo>
      <editorial>Ediciones B</editorial>
    </libro>
  </prestamo>
  <venta>
    <libro isbn="553465201">
      <titulo>Memorias de Idhún I La Resistencia</titulo>
      <autor>Laura Gallego García</autor>
      <anyo>2005</anyo>
      <editorial>Ediciones SM</editorial>
    </libro>
    <libro isbn="173465278">
      <titulo>Next</titulo>
      <autor>Michael Crichton</autor>
      <anyo>2008</anyo>
      <editorial>DeBolsillo</editorial>
    </libro>
  </venta>
</libros>
```

PARTE 1: EJERCICIOS DE LECTURA Y EXPLORACIÓN DE XML (DOM + XPath)

Ejercicio 1 - Lectura básica de un XML con DOM

El objetivo es aprender a leer un XML y mostrar por consola la información de sus nodos.

1. Crea un programa Java que abra el fichero ejercicio1.xml dentro de la carpeta /resources.
2. Muestra por pantalla:
 - El nombre del nodo raíz y cuantos hijos tiene.
 - De cada libro nos interesa conocer el ISBN, el título, autor, año.
 - También nos interesa conocer si está en venta o es para préstamos.

```
Nodo raíz: libros
Total de libros: 4
-----
Sección: prestamo | ISBN: 9734602 | Título: El Ocho | Autor: Katherine Neville | Año: 1988
...
Sección: venta | ISBN: 173465278 | Título: Next | Autor: Michael Crichton | Año: 2008
```



Pista:

- Usa `DocumentBuilder`, `getDocumentElement()`, `getElementsByTagName()` y `getAttribute()`.

Ejercicio 2 - Consultas con XPath

El objetivo es aplicar XPath para buscar información específica dentro del XML.

1. A partir del mismo XML, crea un programa con el siguiente menú:

```
1. Mostrar todo el XML completo
2. Listar los libros prestados
3. Listar los libros en venta
4. Salir
```

2. Usa `XPathFactory` y `XPath.evaluate()` para ejecutar las opciones 2 y 3.
3. Muestra por consola el título, autor y año de cada libro que coincida con la consulta.



Pista:

- Utiliza la clase `XmlCtrlDom` con métodos reutilizables que nos servirá como clase de apoyo para trabajar con documentos XML (cargar, ejecutar consultas XPath y guardar).



Importante: A partir de este ejercicio siempre utilizaremos la clase `XmlCtrlDom.java`

PARTE 2: EJERCICIOS DE CREACIÓN Y MODIFICACIÓN DE XML

Ejercicio 3 - Añadir un nodo nuevo al árbol DOM

El objetivo es aprender a modificar un XML existente.

1. Utiliza el archivo ejercicio1.xml como base.
2. Crea un nuevo nodo <libro> dentro del bloque <prestamo>.

```
<libro isbn="12345678">  
  <titulo>Nuevo libro de ejemplo</titulo>  
  <autor>Autor de prueba</autor>  
  <anyo>2026</anyo>  
  <editorial>Editorial Simarro</editorial>  
</libro>
```

3. Guarda el nuevo XML modificado en ./resources/ejercicio3.xml.



Pista:

- Usa los métodos createElement(), setTextContent() y appendChild().

Ejercicio 4 - Crear un XML desde cero

El objetivo es practicar la generación de un documento XML nuevo.

1. El programa Java deberá generar el siguiente archivo (no se puede copiar el XML a mano):

```
<alumnos>  
  <alumno id="1">  
    <nombre>Juan</nombre>  
    <apellidos>Martí López</apellidos>  
    <edad>18</edad>  
  </alumno>  
  <alumno id="2">  
    <nombre>Amparo</nombre>  
    <apellidos>Luna Dolores</apellidos>  
    <edad>25</edad>  
  </alumno>  
</alumnos>
```

2. Guarda el resultado en ./resources/ejercicio4.xml.



Pista:

- Usa DocumentBuilder.newDocument().

PARTE 3: MAPEOS DE XML A OBJETOS JAVA

Ejercicio 5 - Mapeo de XML a objetos (ArrayList simple)

El objetivo es leer un XML y convertirlo en una lista de objetos Java.

1. Crea una clase Alumno con atributos id, nombre, apellidos, edad.
2. Utiliza el XML creado en el ejercicio anterior con nombre ejercicio4.xml con varios alumnos.
3. El programa debe:
 - Leer el XML.
 - Crear un objeto Alumno por cada <alumno> del XML.
 - Guardarlos en un ArrayList<Alumno>.
 - Mostrarlos por pantalla el ArrayList.



Pista:

- Utiliza DOM para leer los datos y pasarlos a objetos de la clase Alumno.

Ejercicio 6 - Mapeo con separación por curso

El objetivo es separar los alumnos por curso y almacenarlos en distintas listas.

1. Copia el siguiente archivo ejercicio6.xml en la carpeta /resources:

```
<?xml version="1.0" encoding="UTF-8"?>
<alumnos>
  <alumno id="1">
    <nombre>Roberto</nombre>
    <apellidos>Blasco</apellidos>
    <nota>9</nota>
    <curso>2nDAM</curso>
  </alumno>
  <alumno id="2">
    <nombre>Vicente</nombre>
    <apellidos>Martinez</apellidos>
    <nota>9.5</nota>
    <curso>2nASIR</curso>
  </alumno>
  <alumno id="3">
    <nombre>Rosana</nombre>
    <apellidos>Rosello</apellidos>
    <nota>5</nota>
    <curso>2nDAM</curso>
  </alumno>
</alumnos>
```

2. Crea una clase nueva Alumno con los atributos correspondientes a la nueva estructura de datos.
 - La clase que habíamos creado anteriormente no nos sirve porque el XML tiene diferentes atributos.
3. A partir del XML anterior, guarda los alumnos en dos listas diferentes: una para los alumnos de 2nDAM y otra para los de 2nASIR.
4. Muestra por pantalla cada lista por separado con el nombre del curso y sus alumnos.



Pista:

- Se han de crear dos ArrayList: claseDAM y claseASIR.

Ejercicio 7 - Mapeo con ordenación por nota

El objetivo es aplicar Comparable para ordenar objetos según su nota media.

1. Trabajaremos con el campo nota en el XML y en la clase Alumno.
2. Implementa Comparable<Alumno> para ordenar por nota de mayor a menor.
 - Por lo tanto, hemos de modificar la clase Alumno creada en el anterior ejercicio.
3. Muestra por consola los alumnos ordenados por nota para cada curso.
 - Puedes añadir alumnos de forma manual para tener más alumnos en el archivo. Podéis decir a cualquier IA que os añada a 5 alumnos de 2nASIR y 5 de 2nDAM.



Pista:

- Utilizar la **interfaz Comparable** de Java, que podéis ver un ejemplo en el siguiente enlace:
 - <https://www.arquitecturajava.com/java-comparable-interface-y-ordenaciones/>