

# XPATH

¿Te imaginas tener que trabajar con DOM todo el rato? Hay que reconocer que DOM tiene su utilidad, pero cuando estamos hablando de ficheros XML de tamaño y complejidad considerable, deja de ser práctico. Sobre todo, cuando quieres hacer una búsqueda de información. Y en programación cuando algo se complica demasiado, se inventa algo para hacerlo más fácil.

**XPath** o XML Path es un lenguaje que permite navegar y seleccionar los nodos de un documento XML de forma razonablemente similar a como haríamos una select en base de datos. Se utiliza tanto para manipular ficheros XML como para páginas web. La idea es parecida a las expresiones regulares para seleccionar partes de un texto plano. XPath permite buscar y seleccionar nodos respetando la estructura jerárquica. Fue originalmente creado par su uso con el estándar XSLT, y definido por la W3C. Puedes mirar este [enlace](#) para descubrir todas las posibilidades del lenguaje.

## XPATH

Java tiene integrado XPath mediante un conjunto de clases en el paquete **javax.xml.xpath**. La clase **XPathFactory** se genera una instancia de XPath con la que podremos crear un objeto **XPathExpression** para realizar las consultas.

Las consultas con XPath son en realidad unas cadenas con la estructura de nodos a buscar separadas por "/". Por ejemplo, para este caso:

```
<?xml version="1.0" encoding="UTF-8"?>
<students>
  <student id="1">
    <name>Juan</name>
    <surname>Torres</surname>
    <signature>ADT</signature>
  </student>
  <student id="2">
    <name>Ana</name>
    <surname>Sanz</surname>
    <signature>PMDM</signature>
  </student>
</students>
```

Podríamos hacer las siguientes búsquedas:

**/students/student:** Devuelve el Nodo < student > con todos sus nodos hijos.

**/students/student/name:** Devuelve toda la lista de nodos <name> que estén dentro de student.

**/students /descendant::name:** Devuelve toda la lista de nodos <name> que sean hijos, nietos, bisnietos, etc. de <students>, sin importar dónde estén.

**Count(/students/student):** Cuenta los <student> que hay en <students>

**/students/descendant::student[@id="1"]:** Similar a un 'where' de SQL, se recupera el nodo <student> que tiene el atributo id = 1.

Por ejemplo, para buscar y listar todos los estudiantes:

```
// Get the Factory
factory = DocumentBuilderFactory.newInstance();

// Get the DocumentBuilder
builder = factory.newDocumentBuilder();

// Parse the XML file into a Document
Document xPathDoc = builder.parse(new File("XPathExample.xml"));

// The XPath
XPath xPath = XPathFactory.newInstance().newXPath();

// The expression ("query")
String expression = "/students/student";

// Searching...
NodeList nodeList = (NodeList) xPath.evaluate(expression, xPathDoc, XPathConstants.NODESET);

// Let's iterate
for (int i = 0; i < nodeList.getLength(); i++) {
    Node node = nodeList.item(i);
    if (node.getNodeType() == Node.ELEMENT_NODE) {
        Element element = (Element) node;
        System.out.println("student: " + i);
        System.out.println("id: " + element.getAttribute("id"));

        NodeList childNodes = node.getChildNodes();
        for (int j = 0; j < childNodes.getLength(); j++) {
            Node childNode = childNodes.item(j);
            if (childNode.getNodeType() == Node.ELEMENT_NODE) {
                Element childElement = (Element) childNode;
                System.out.println("childElement: " + childElement.getTextContent());
            }
        }
    }
}
```

## Práctica - 5

---

Partiendo del XML generado en la práctica anterior, modifica el programa para añadir las siguientes consultas a colección de videojuegos.

```
Videojuego: DOOM3
Distribuidor: Activision
Fecha: 2007
Tipo: FPS
Jugadores: 1
```

- Buscar por nombre: Busca un videojuego por nombre y lo muestra.
- Buscar por fecha: Busca un videojuego por fecha y lo muestra.
- Numero Juegos: Muestra el número de juegos totales.

## Práctica - 6

---

Diseña un XML que permita gestionar todos los alumnos del centro. En el centro hay dos ciclos, DAM y DAW, pero para cada uno hay una clase de primero y una de segundo (cuatro clases en total). Cada alumno sólo puede pertenecer a una clase, pero es posible que haya alumnos con el mismo nombre.

Realiza las siguientes consultas:

- Muestra todos los alumnos del centro
- Muestra todos los alumnos de DAM
- Muestra todos los alumnos de DAW
- Muestra todos los alumnos de 1º y de 2º de cada ciclo (4 consultas).
- Muestra a un alumno por nombre
- Muestra un alumno por nombre de un ciclo y curso concreto
- Número total de alumnos de cada ciclo