

## Anexo: Expresiones Xpath en Java

# 1. Ejecutar expresiones XPATH

### Paso 1: Cargar el documento XML

Para cargar un documento XML, primero se crea una instancia de DocumentBuilderFactory y DocumentBuilder. Luego, se parsea el archivo XML.

```
DocumentBuilderFactory fabrica = DocumentBuilderFactory.newInstance();
DocumentBuilder procesador = fabrica.newDocumentBuilder();
Document documento = procesador.parse("documento.xml");
```

### Paso 2: Crear una instancia de XPathFactory y Xpath

Se crea una instancia de **XPathFactory** que te permitirá crear objetos **XPath** para evaluar las expresiones XPath.

```
XPathFactory xpathFabrica = XPathFactory.newInstance();
XPath xpath = xpathFabrica.newXPath();
// o también
XPath xpath = XPathFactory.newInstance().newXPath();
```

### Paso 3: Compilar y evaluar una expresión XPath

Utiliza la instancia de XPath para **compilar y evaluar** expresiones XPath en el documento XML.

```
XPathExpression expresion = xpath.compile("expresion XPath ");
NodeList resultado = (NodeList) expresion.evaluate(documento,
                                                     XPathConstants.NODESET)
// aquí el resultado se espera un conjunto de nodos

// o también directamente sin tener que compilar la expresión
NodeList resultado = (NodeList) xpath.evaluate("expresion XPath", documento,
                                                XPathConstants.NODESET);
```

**XPathConstants:** es una clase que **proporciona constantes que te permiten especificar el tipo de resultado que se espera** al ejecutar expresiones XPath en documentos XML, lo que facilita el manejo de los resultados de manera adecuada. Opciones principales en XPathConstants:

- **XPathConstants.NODE:** Esta constante se utiliza para especificar que se espera obtener un solo nodo como resultado de la evaluación de la expresión XPath. El resultado se almacena como un objeto *org.w3c.dom.Node*.
- **XPathConstants.NODESET:** Usado cuando esperas un conjunto de nodos como resultado. El resultado se almacena en un objeto *org.w3c.dom.NodeList*.
- **XPathConstants.STRING:** Esta constante se usa cuando se espera una cadena como resultado. El resultado se almacena como un objeto *java.lang.String*.
- **XPathConstants.NUMBER:** Utilizado cuando esperas un valor numérico (decimal) como resultado. El resultado se almacena como un objeto *java.lang.Double*.
- **XPathConstants.BOOLEAN:** Utilizado cuando esperas un valor booleano como resultado. El resultado se almacena como un objeto *java.lang.Boolean*.

Por ejemplo, si deseas obtener el contenido de un elemento como una cadena, usarías

XPathConstants.STRING de la siguiente manera:

```
XPathExpression expresion = xpath.compile(xpathExpresion);
String resultado = (String) expresion.evaluate(documento,
                                              XPathConstants.STRING);
```

Ejemplo: Dado el siguiente Documento en memoria

```
<biblioteca>
  <seccion nombre="Programación" estanteria="A1">
    <libros>
      <libro codigo="001">
        <titulo>Estructuras de Datos en C++</titulo>
        <autores>
          <autor>Mark Allen Weiss</autor>
          <autor>Robert Sedgewick</autor>
        </autores>
        <editorial>Pearson</editorial>
        <ejemplares>
          <ejemplar numero="1" estado="Disponible"/>
          <ejemplar numero="2" estado="Prestado"/>
        </ejemplares>
      </libro>
      <libro codigo="002">
        <titulo>Python for Data Science</titulo>
        <autores>
          <autor>Wes McKinney</autor>
        </autores>
        <editorial>O'Reilly</editorial>
        <ejemplares>
          <ejemplar numero="1" estado="Disponible"/>
        </ejemplares>
      </libro>
    </libros>
  </seccion>
  <seccion nombre="Redes y Seguridad" estanteria="B2">
    <libros>
      <libro codigo="003">
        <titulo>Computer Networking: Principles, Protocols, and Practice</titulo>
        <autores>
          <autor>Olivier Bonaventure</autor>
        </autores>
        <editorial>self-published</editorial>
        <ejemplares>
          <ejemplar numero="1" estado="Disponible"/>
          <ejemplar numero="2" estado="Reservado"/>
          <ejemplar numero="3" estado="Dañado"/>
        </ejemplares>
      </libro>
    </libros>
  </seccion>
  <seccion nombre="Inteligencia Artificial" estanteria="C3">
    <libros>
      <libro codigo="004">
        <titulo>Artificial Intelligence: A Modern Approach</titulo>
        <autores>
          <autor>Stuart Russell</autor>
          <autor>Peter Norvig</autor>
        </autores>
        <editorial>Pearson</editorial>
        <ejemplares>
          <ejemplar numero="1" estado="Disponible"/>
          <ejemplar numero="2" estado="Prestado"/>
          <ejemplar numero="3" estado="Disponible"/>
        </ejemplares>
      </libro>
    </libros>
  </seccion>
```

```

</ejemplares>
</libro>
</libros>
</seccion>
</biblioteca>

```

1. Seleccionar todos los ejemplares con estado "Dañado" en la biblioteca:

```

XPathExpression expresion = xpath.compile("//ejemplar[@estado='Dañado']");
NodeList ejemplaresDanados = (NodeList) expression.evaluate(documento,
XPathConstants.NODESET);

```

2. Calcular cuántos libros hay en la biblioteca:

```

XPathExpression expression = xpath.compile("count(/libro)");
Double cantidadLibros = (Double) expression.evaluate(documento,
XPathConstants.NUMBER);
int cantidadLibrosEntero = cantidadLibros.intValue();

```

3. Obtener el título del primer libro de la sección Redes y seguridad:

```

XPathExpression expression = xpath.compile
("/biblioteca/seccion[@nombre='Redes y Seguridad']/libros/libro[1]/titulo");
String tituloPrimerLibroRedesSeguridad = (String) expression.evaluate
(documento, XPathConstants.STRING);

```

4. Obtener el libro con código 004:

```

XPathExpression expression = xpath.compile("//libro[@codigo='004']");
Node libro = (Node) expression.evaluate(documento, XPathConstants.NODE);

```

5. Verificar si hay libros prestados :

```

XPathExpression expression = xpath.compile
("boolean(//ejemplar[@estado='Prestado'])");
boolean hayEjemplaresPrestados = (Boolean) expression.evaluate(documento,
XPathConstants.BOOLEAN);

```