
LENGUAJE DE MARCAS

C.F.G.S.: ASIR

XPATH

1. INTRODUCCIÓN

Se usa para navegar por los elementos y atributos de un documento XML.

Se utiliza en XSLT (Lo vemos más adelante).

Existen tres versiones de XPath aprobadas por el W3C, aunque la versión más utilizada sigue siendo la versión 1:

- noviembre de 1999: [XML Path Language 1.0](#)
- enero de 2007: [XML Path Language 2.0](#)
- diciembre de 2010: [XML Path Language 2.0 \(2ª edición\)](#)
- abril de 2014: [XML Path Language 3.0](#)

2. NODOS

Los documentos XML son tratados como árboles de nodos.

En el siguiente ejemplo:

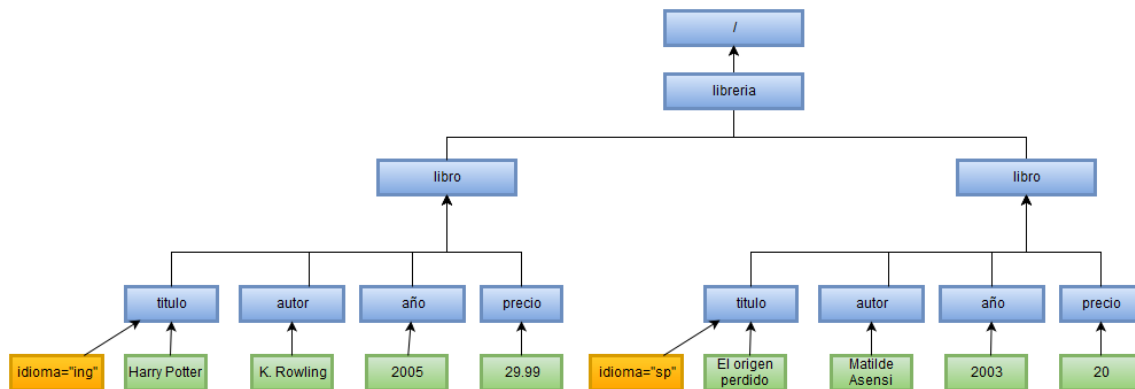
```
<libreria>

  <libro>
    <titulo idioma="ing">Harry Potter</titulo>
    <autor>J K. Rowling</autor>
    <año>2005</año>
    <precio>29.99</precio>
  </libro>

  <libro>
    <titulo idioma="sp">El origen perdido</titulo>
    <autor>Matilde Asensi</autor>
    <año>2003</año>
    <precio>20</precio>
  </libro>

</libreria>
```

El árbol de nodos que se crearía:



Las relaciones entre los nodos serían:

Parent: Cada elemento o atributo tiene un parent.

Ej. El elemento libro es el parent de titulo, autor, año y precio.

Children: Los nodos pueden tener uno, 0 o muchos hijos.

Ej. Los elementos titulo, autor, año y precio son children de libro.

Siblings: Son nodos que tienen el mismo parent.

Ej. titulo, autor, año y precio son siblings.

Ancestors: El parent, el parent del parent, etc.

Ej. Los ancestors de titulo son libro y libreria.

Descendants: Children, los children de los children, etc...

Ej. Los descendants de libreria son libro, titulo, autor, año y precio.

En XPath no sólo los elementos son nodos, en realidad hay siete tipos de nodos:

- **Raíz:** Se identifica por /. Hace referencia al nodo raíz del árbol, pero no del documento xml. En el ejemplo el nodo raíz del árbol contiene al elemento raíz libreria.
- **Elemento:** Etiqueta. Su padre es otro elemento, excepto el elemento raíz cuyo padre es el nodo raíz. Pueden tener hijos que pueden ser elementos, texto, comentario, instrucción de procesamiento. Pueden tener propiedades como atributos o espacios de nombres.
- **Atributo**
- **Texto:** Caracteres fuera de las etiquetas.
- **Comentario**
- **Instrucción de procesamiento** (más adelante)
- **Espacio de nombres**

3. SINTAXIS DE LAS EXPRESIONES XPATH

Una expresión XPath es una cadena de texto que representa un recorrido en el árbol del documento. Las expresiones más simples se parecen a las rutas de los archivos en Linux.

Evaluar una expresión XPath es buscar si hay nodos en el documento que se ajustan al recorrido definido en la expresión. El resultado de la evaluación son todos los nodos que se ajustan a la expresión. Para poder evaluar una expresión XPath, el documento debe estar bien formado.

Las expresiones XPath se pueden dividir en pasos de búsqueda. Cada paso de búsqueda se puede a su vez dividir en tres partes:

- eje: selecciona nodos elemento o atributo basándose en sus nombres.
- predicado: restringe la selección del eje a que los nodos cumplan ciertas condiciones.
- selección de nodos: de los nodos seleccionados por el eje y predicado, selecciona los elementos o el texto que contienen o ambos.

Vemos como funcionan las distintas expresiones xpath sobre el ejemplo anterior:

EJE (Hacha)

Los nodo elemento se indican con el nombre del elemento y los nodo atributo con @ y el nombre del atributo.

/	Si está al principio de la expresión indica el nodo raíz, sino indica "hijo". Debe ir seguida del nombre de un elemento
---	---

/libreria/libro/autor

```
<autor>J K. Rowling</autor>
<autor>Matilde Asensi</autor>
```

/autor

No devuelve nada ya que autor no es hijo del nodo raíz

/libreria/autor

No devuelve nada ya que autor no es hijo de librería

/libreria/libro/titulo/@idioma

idioma="ing"

idioma="sp"

/librería/libro/@idioma

No devuelve nada porque el elemento libro no tiene ningún atributo idioma.

//	Indica descendiente (hijos, hijos de hijos, etc...)
----	---

/librería//autor

<autor>J K. Rowling</autor>

<autor>Matilde Asensi</autor>

//autor

<autor>J K. Rowling</autor>

<autor>Matilde Asensi</autor>

//autor//libro

No devuelve ninguno porque libro no es descendiente de autor

//@idioma

idioma="ing"

idioma="sp"

/..	Indica el elemento padre
-----	--------------------------

/librería/libro/título/@idioma/..

<título idioma="ing">Harry Potter</título>

<título idioma="sp">El origen perdido</título>

//@idioma/..../..

<libro>

<título idioma="ing">Harry Potter</título>

<autor>J K. Rowling</autor>

<año>2005</año>

<precio>29.99</precio>

</libro>

<libro>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

<año>2003</año>

<precio>20</precio>

</libro>

	Permite indicar varios recorridos
--	-----------------------------------

//autor | //titulo

<titulo idioma="ing">Harry Potter</titulo>

<autor>J K. Rowling</autor>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

PREDICADO

Se escribe entre corchetes a continuación del eje. El predicado restringe los nodos a los que cumplen unas determinadas condiciones.

[@atributo]	Selecciona los elementos que tienen ese atributo
-------------	--

//titulo[@idioma]

<titulo idioma="ing">Harry Potter</titulo>

<titulo idioma="sp">El origen perdido</titulo>

[numero]	Si hay varios resultados selecciona uno de ellos por orden, last() selecciona el último
----------	---

//libro[1]

<libro>

<titulo idioma="ing">Harry Potter</titulo>

<autor>J K. Rowling</autor>

<año>2005</año>

<precio>29.99</precio>

</libro>

//libro[last()]

<libro>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

<año>2003</año>

<precio>20</precio>

</libro>

[condición]	Selecciona los nodos que cumplen la condición.
-------------	--

En la condición se pueden utilizar los siguientes operadores:

- Lógicos: and, or, not()
- Aritméticos: +, -, *, div, mod
- Comparación: =, !=, <, >, <=, >=

Si se compara con cadenas de texto éstas deben ir entre comillas simples o dobles.

//titulo[@idioma="sp"]

<titulo idioma="sp">El origen perdido</titulo>

//libro[autor="Matilde Asensi"]

<libro>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

<año>2003</año>

<precio>20</precio>

</libro>

//libro[precio>25]

<libro>

<titulo idioma="ing">Harry Potter</titulo>

<autor>J K. Rowling</autor>

<año>2005</año>

<precio>29.99</precio>

</libro>

//autor[.="Matilde Asensi"]

<autor>Matilde Asensi</autor>

Más ejemplos de predicados:

/libreria/libro[last()-1]	Penúltimo elemento libro hijo del elemento libreria
/libreria/libro[position()<3]	Dos primeros elementos libro del elemento libreria
/libreria/libro[precio>35.00]/titulo	Todos los elementos titulo de los elementos libro que tienen un precio mayor de 35.00
//titulo[@*]	Todos los elementos titulo que tienen cualquier atributo

SELECCIÓN DE NODOS

/node()	Selecciona todos los hijos (elementos o texto del nodo)
---------	---

//libro/node()

<titulo idioma="ing">Harry Potter</titulo>

<autor>J K. Rowling</autor>

<año>2005</año>

<precio>29.99</precio>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

<año>2003</año>

<precio>20</precio>

//autor/node()

J K. Rowling

Matilde Asensi

/text()	Selecciona únicamente el texto contenido en el nodo
---------	---

//autor/text()

J K. Rowling

Matilde Asensi

//libro/text

No muestra nada porque el elemento libro no tiene texto.

/*	Selecciona todos los hijos, sólo elementos, del nodo
----	--

//libro/*

<titulo idioma="ing">Harry Potter</titulo>

<autor>J K. Rowling</autor>

<año>2005</año>

<precio>29.99</precio>

<titulo idioma="sp">El origen perdido</titulo>

<autor>Matilde Asensi</autor>

<año>2003</año>

<precio>20</precio>

//@*	Selecciona todos los atributos del nodo
------	---

//titulo/@*

idioma="ing"

idioma="sp"

4. SINTAXIS COMPLEJA DE LAS EXPRESIONES XPATH (No lo vemos)

Las expresiones que hemos visto en el apartado anterior se pueden expresar de forma más compleja.

La sintaxis es uno o más pasos con la forma definida abajo y separados por / :

nombredeleje::nombredelnodo[predicado]

Los distintos nombres de eje son:

Nombre del eje	Resultado
Ancestor	Selecciona todos los antecesores (padre, abuelo, etc.) del nodo actual
ancestor-or-self	Igual que el anterior y además el propio nodo
Attribute	Selecciona todos los atributos del nodo actual
Child	Selecciona todos los hijos del nodo actual.
Descendant	Selecciona todos los descendientes (hijos, nietos, etc.) del nodo actual.
descendant-or-self	Igual que el anterior pero añadiendo el propio nodo.
Following	Selecciona todo en el documento después del tag de cierre del nodo actual.
following-sibling	Selecciona todos los hermanos después del nodo actual
Namespace	Selecciona todos los nodos de espacios de nombres de el nodo actual
Parent	Selecciona el padre del nodo actual.
Preceding	Selecciona todo lo que está antes del tag de apertura del nodo actual
preceding-sibling	Selecciona todos los hermanos antes del nodo actual
Self	Selecciona el nodo actual.

Hay algunas equivalencias con las expresiones que hemos utilizado en el punto anterior:

/child:: equivale a /
 attribute:: equivale a @
 /descendant:: equivale a //

Ejemplos	Resultado
child::libro	Selecciona todos los nodos libro que son children del nodo actual
attribute::idioma	Selecciona el atributo idioma del nodo actual
child::*	Selecciona todos los children del nodo actual
attribute::*	Selecciona todos los atributos el nodo actual

child::text()	Selecciona todos los nodos de texto children del nodo actual
child::node()	Selecciona todos los nodos children del nodo actual
descendant::libro	Selecciona todos los libros descendientes de el nodo actual
ancestor::libro	Selecciona todos los libros ancestors del nodo actual
ancestor-or-self::libro	Selecciona todos los libros ancestors del nodo actual y el nodo actual si también es un nodo.
child::* / child::precio	Selecciona todos los precios nietos del nodo actual.

/child::libreria/child::libro es igual a /librería/libro

/descendant::libro es igual a //libro

/descendant::titulo/attribute::idioma es igual a //titulo/@idioma

Esta sintaxis compleja nos permite también crear algunas expresiones xpath que no se podrían crear con la sintaxis más sencilla.

Bibliografía:

http://www.mclibre.org/consultar/xml/lecciones/xml_xpath.html

www.w3schools.com