XPath

Juan Ignacio Raggio

May 3, 2025

Contents

1 Consultas en documentos XML

- XPath
  - XPath engine puede ser interesante de implementar
- XQuery
- XSLT

2 Como funciona?

- Se tienen paths para identificar y navegar a travez de los nodos de un documento xml
- La forma de guardar los paths es en forma de arbol, representados respetando la jerarquia dada
- Vamos a ignorar los namespaces y headers (tomamos como nodo raiz el tag raiz del documento)

3 Exppresiones XPath:

Tienen el siguiente aspecto: *step1/step2...*

- Steps
  - Axis
  - Node test
  - Cero o mas predicados para restringir nodos

4 Forma de los steps:

step = axis::nodeTest [predicado]

Axis para posicionamiento	Semántica: Son los mostrados en la figura. Define un conjunto de nodos relativos al actual
self::	Selecciona el nodo contexto actual.
child::	Selecciona hijos del nodo contexto actual, cuyo nodeTest dé true. Es la que se aplica cuando no se indica axis (por default).
descendant::	Selecciona nodos descendientes del nodo contexto actual (recursivamente, no sólo en un nivel) cuyo nodeTest dé true.
descendant-or-self::	Selecciona el nodo contexto actual y todos sus descendientes cuyo nodeTest dé true.
ancestor::	Selecciona los nodos ancestros del nodo contexto actual (hasta llegar a la raíz) cuyo nodeTest dé true.
ancestor-or-self::	Selecciona el nodo contexto actual y sus nodos ancestros cuyo nodeTest dé true.
following::, following-sibling::, preceding::, etc	Ver figura del próximo slide.
attribute::	Selecciona los atributos del nodo contexto actual cuyo nodeTest dé true.

Node Test	Semántica: los siguientes son los posibles node test a utilizar
QName	Devuelve true si el nodo se llama QName. Ej: child::curso o attribute::ID
@QName	Devuelve true si el nodo atributo se llama QName. Ej: @ID en vez de attribute::ID
*	Wildcard. Devuelve true siempre, no importa el nombre. Usado en /child::* selecciona todos los nodos hijos del nodo contexto. Usado en /attribute::* selecciona todos los nodos atributos del nodo contexto. No sirve para seleccionar nodo texto.
node()	Devuelve true para el nodo en cuestión. Parecido al wildcard *, pero sólo da true si se está en un nodo elemento (no sirve para atributo).
text()	Devuelve true si el nodo es de tipo texto.

4.1 Como seleccionar nodos hijos?

/child::A -> donde `A` es el nombre de un nodo hijo (selecciona la lista de nodos hijos con ese nombre) /A -> Abreviatura valida para seleccionar nodos hijos con nombre `A`

4.2 Como seleccionar atributos?

/attribute::A -> Donde `A` es el nombre del atributo /@A -> Abreviatura valida para seleccionar atributos

4.3 Como seleccionar recursivamente todos los nodos hijos de un nodo?

/descendant::A -> Selecciona a `todos` los nodos hijos de `A` (recursivamente)
OJO: /descendant::. . . no selecciona attributos del nodo contexto actual

- En caso de que quisieramos incluir los nodos atributo del contexto actual podemos usar: /descendant-or-self::A //A -> Abreviatura para `todos` los nodos hijos de `A` (recursivamente)

4.4 Nodos predecesores

/ancestor::A -> Selecciona como descendant (recursivamente) hacia arriba

Como en descendant, no podemos seleccionar nodos attribute del nodo contexto actual, tenemos que usar: /ancestor-or-self::A

4.5 Nodos parent

Seleccionar los nodos hijos parent (un nivel hacia arriba, es decir: `antecesor inmediato`) llamados `A` de una lista de nodos contexto *parent::A -> Subi un nivel seleccionando a los padres de los nodos contexto actual /parent::* -> Subi un nivel, no me importa el nombre de mi padre .. ->* Abreviatura

4.6 Nodos contexto

/self::A -> Filtra la seleccion actual a que se quede con los nodos llamados `A`

4.7 Predicados

Permite filtrar sobre la seleccion actual

- Los paths dentro de un predicado son independientes del path "real"

5 Ejemplo:

```
<?xml version="1.0"?>
<?mso-application progid="Word.Document"?>
<!-Este es el xml del ejemplo visto en clase -->
<universidad xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance"
             xsi:noNamespaceSchemaLocation="uni.xsd">
  <curso ID="XML">
```

```
    <alumno ID="_100">
      <nota>7</nota>
    </alumno>
    <alumno ID="_200">
      <nota>3</nota>
      <nota>5</nota>
    </alumno>
  </curso>
  <curso ID="OOP">
    <alumno ID="_200">
      <nota>9</nota>
    </alumno>
    <alumno ID="_400">
      <nota>3</nota>
      <nota>2</nota>
    </alumno>
  </curso>
  <alumno ID="_100">
    <name>Ana</name>
  </alumno>
  <alumno ID="_200">
    <name>Salerno</name>
  </alumno>
  <alumno ID="_400"/>
</universidad>
```

5.1 Ejemplo 1:

/child::universidad/child::alumno

1. Seleccionamos nodos hijos de universidad
2. Seleccionamos nodos "hijos" alumno

Esto selecciona todos los nodos hijos de ese path

5.2 Ejemplo 2:

/child::universidad/child::curso/attribute::ID

1. Seleccionamos a los hijos de `universidad`
2. Seleccionamos a los hijos de `cursos` en el contexto de universidad

3. Seleccionamis al hijo `ID` de los nodos `attribute` en el contexto dado (universidad -> curso) (si hay mas de un nodo curso lo buscamos en los dos)

Notemos que la expresion para seleccionar un nodo hoja es distinto

5.3 Ejemplo Descendant:

/child::universidad/descendant::*/attribute::ID

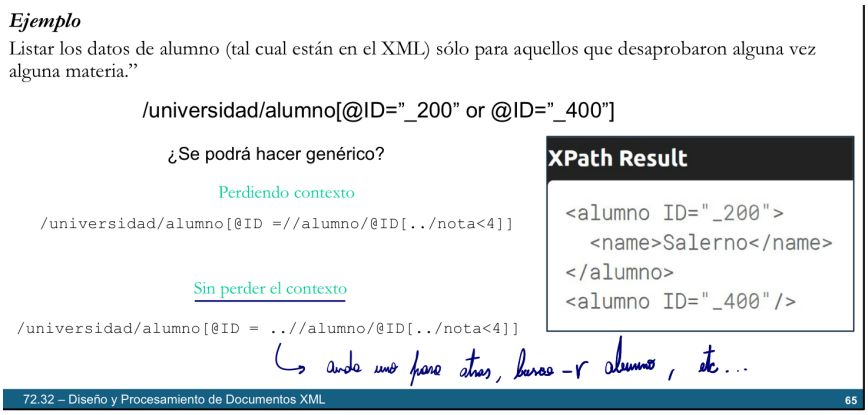
1. Sel hijos de universidad
2. Sel todos los hijos recursivamente en el contexto dado
3. Seleccionamos todos los nodos de tipo @ID (attribute::ID) dentro de los nodos seleccionados

5.4 Ejemplo predicado:

- Todos los ID de alumnos que desaprobaron alguna vez alguna materia

//alumno/@ID[../nota<4]

5.5 Ejemplo de perder contexto vs no perder contexto en predicados:



6 Funciones

6.1 Funciones numericas

- Las conversiones de String a numero se hacen automaticamente, pero es `recomendable hacer el casteo explicito`

Sintaxis	Semántica
ceiling(number**)**	Devuelve el menor número entero que no sea menor al argumento. Ej: ceiling(3.2) es 4 Ej: ceiling(3.7) es 4
floor(number**)**	Devuelve el menor número entero que no sea mayor al argumento. Ej: floor(3.2) es 3 Ej: floor(3.7) es 3
round(number**)**	Devuelve el número entero más próximo al argumento. Si hubiera dos, el más cercano al infinito positivo. Ej: round(3.2) es 3 Ej: round(3.7) es 4
sum(node-set**)**	Realiza la sumatoria de los valores.
number(string**)**	Convierte si se puede el número que está en el string subyacente a tipo number. Ej: number("3.2") es 3.2

Sintaxis para operadores numéricos	Semántica
nro1 **div** nro2	Calcula la división entera entre nro1 y nro2
nro1 **mod** nro2	Calcula el resto de la división entera entre nro1 y nro2
+,-,*	Típicos operadores aritméticos

6.2 Funciones de strings XPath

Sintaxis para funciones sobre strings	Semántica: Tener en cuenta que un string no puede contener por ejemplo el símbolo <, y debería usarse &lt, etc
starts-with(string1, string2**)**	Devuelve true si string1 comienza con string2.
contains(string1, string2**)**	Devuelve true si string1 contiene a string2.
string-length(string**)**	Devuelve la cantidad de caracteres del string.
concat(string1, string2, string***)**	Devuelve un nuevo string concatenando los argumentos (por lo menos 2).
substring-before(string1, string2**)**	Devuelve el substring de string1 que aparece antes que string2. Ej: substring-before('acbcbch', 'bc') devuelve 'ac'.
substring-after(string1, string2**)**	Devuelve el substring de string1 que aparece después que string2. Ej: substring-after('acbcbch', 'bc') devuelve 'bch'.
substring(string, posInicial, longitud**)**	El segundo argumento se interpreta como índice inicial (la numeración comienza en 1) y el tercero como longitud solicitada. Devuelve el substring del primer argumento que comienza en el índice especificado y de longitud pedida. Ej: substring('Documento XML', 11, 5) devuelve 'XML' , porque (en vez de 5 caracteres pudo tomar sólo 3. Ej: substring('Documento XML', -1, 4) devuelve 'Do' porque la posición -1 y 0 no son usadas porque están fuera de los límites (el índice comienza en 1) y la longitud queda en 2 ya que los primeros 2 caracteres no estaban dentro de los límites.

6.3 Funciones de node-set

Sintaxis para funciones sobre node-set	Semántica
count(node-set**)**	Devuelve la cantidad de nodos del argumento. Ya lo usamos en ejemplos previos.
name()	Devuelve el nombre del atributo o elemento.
last()	Devuelve el último nodo de un conjunto. **Se ejecuta ANTES de establecer contexto.**
position()	Devuelve la posición del nodo. **Se ejecuta ANTES de establecer contexto.**
text()	Retorna el contenido texto del elemento al que se aplique.

6.4 Funciones logicas

Sintaxis para funciones booleanas	Semántica
not(boolean**)**	Devuelve la negación del argumento.

Sintaxis para operadores booleanos	Semántica
or	Clásico.
and	Clásico.

Sintaxis para operadores booleanos	Semántica
=	Compara por equivalencia.
!=	Compara por no equivalencia.
<	Compara por menor.
<=	Compara por menor o igual.
>	Compara por mayor.
>=	Compara por mayor o igual.

6.4.1 Ejemplo de uso: Aquellos alumnos hijos de universidad a los cuales no se les cargo el name:

```
/universidad/alumno[not(name)]
```

7 Combinacion de node-sets

Sintaxis para operador de combinación	Semántica
nodeSet1 | nodeSet2	Combina 2 conjunto de nodos en uno solo.

Ejemplo

Obtener aquellos elementos o aquellos atributos cuyos nombres contengan ID, no importa dónde estén

```
//*[contains(name(),'ID')] | //attribute::*[contains(name(),'ID') ]
```

```
<universityID>
    <curso IDENTIFIER="XML" cupo='20'>
        <alumno ID="100" name='Ana'>
            <nota>7</nota>
        </alumno>
        <alumno ID="200" name='Salerno'>
            <nota>3</nota>
            <nota>5</nota>
        </alumno>
    </curso>
</universityID>
```

XPath Result

```
<universityID>
  <curso IDENTIFIER="XML" cupo="20">
    <alumno ID="100" name="Ana">
      <nota>7</nota>
    </alumno>
    <alumno ID="200" name="Salerno">
      <nota>3</nota>
      <nota>5</nota>
    </alumno>
  </curso>
</universityID>
IDENTIFIER="XML"
ID="100"
ID="200"
```