

UT5. Conversión y Adaptación de Documentos XML

<xpath>

ÍNDICE

1. ¿Qué es XPath?	3
1.2 Árbol de Nodos	3
1.2.1 Tipos de nodos	4
1.3 Sintaxis de las expresiones XPath	5
1.4 Sintaxis abreviada.....	6
1.4.1 Eje	6
1.4.2 Predicado	9
1.4.3 Selección de nodos	12
1.5 Pasos de búsqueda consecutivos.....	16
1.6 Expresiones anidadas.....	18
2. XSL: Lenguaje de Hojas de Estilo Extensible	19
2.1 XSLT	20
3. Referencias:	20

Los documentos XML son documentos de texto con etiquetas que contienen **exclusivamente información sin entrar en detalles de formato**. Esto implica la **necesidad de algún medio para extraer la información de un documento XML**, para que una persona pueda utilizar directamente los datos para leer, imprimir, etc. **A esto se le conoce como transformación de documentos**.

Las tecnologías que entran en juego en la transformación de documentos son:

- XSLT: permite definir el modo de transformar un documento XML en otro.
- XSL-FO: Se utiliza para transformar XML en un formato legible e imprimible por una persona, por ejemplo, en un documento PDF.
- Xpath: permite el acceso a los diversos componentes de un documento XML

1. ¿Qué es XPath?

XPath (XML Path Language) es un lenguaje que permite recuperar información de un documento XML.

Para ello **define una sintaxis para establecer partes** en un documento XML, **permitiendo navegar a través de sus elementos y atributos**, además permite manipular de forma básica booleanos, números y cadenas.

Utiliza un tipo de notación similar a las rutas de los ficheros, pero haciendo referencia a los nodos de un XML.

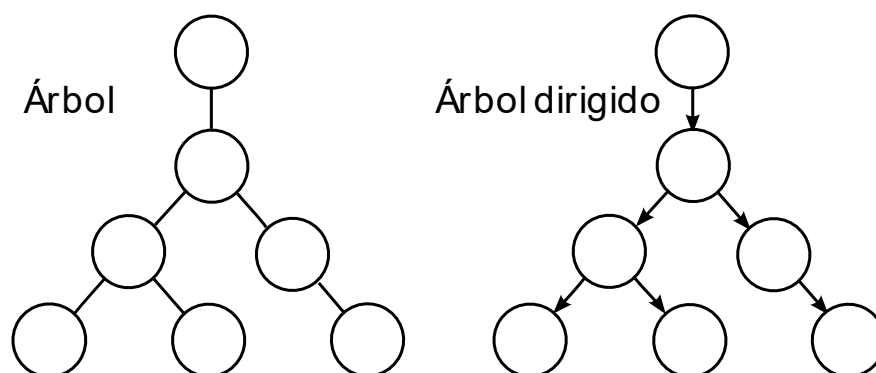
Existen diversas aplicaciones en las que se puede trabajar con XPath para su desarrollo o pruebas, algunas son:

- [XML Copy Editor](#)
- Extension Visual Studio Code: [XPath Tester](#)

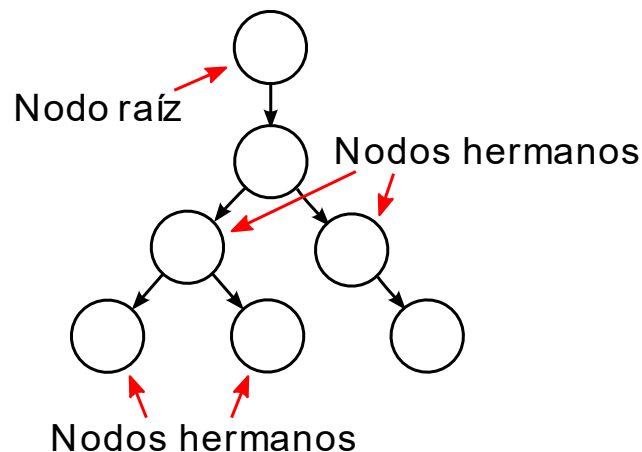
1.2 Árbol de Nodos

XPath considera un documento XML como un árbol de nodos. En Informática, un árbol es una estructura de datos que equivale a un árbol matemático. En Matemáticas un árbol es un caso particular de grafo

Un árbol de nodos se genera tras procesar un documento XML mediante un parser o analizador y se puede representar gráficamente de la siguiente forma:



En XML, existe un único **nodo raíz** de un árbol de nodos. Los **nodos hermanos** son los nodos que tienen el mismo padre.



1.2.1 Tipos de nodos

Un documento XML puede representarse como un árbol dirigido, considerando por ejemplo **los elementos como nodos** y que **un elemento es padre de los elementos que contiene**. Pero en XPath no sólo los elementos son nodos, en realidad hay siete tipos de nodos:

- Raíz
- Elemento
- Atributo
- Texto
- Comentario
- Instrucción de procesamiento
- Espacio de nombres

Nota: La declaración DOCTYPE no se considera como nodo.

Por ejemplo, el documento XML siguiente:

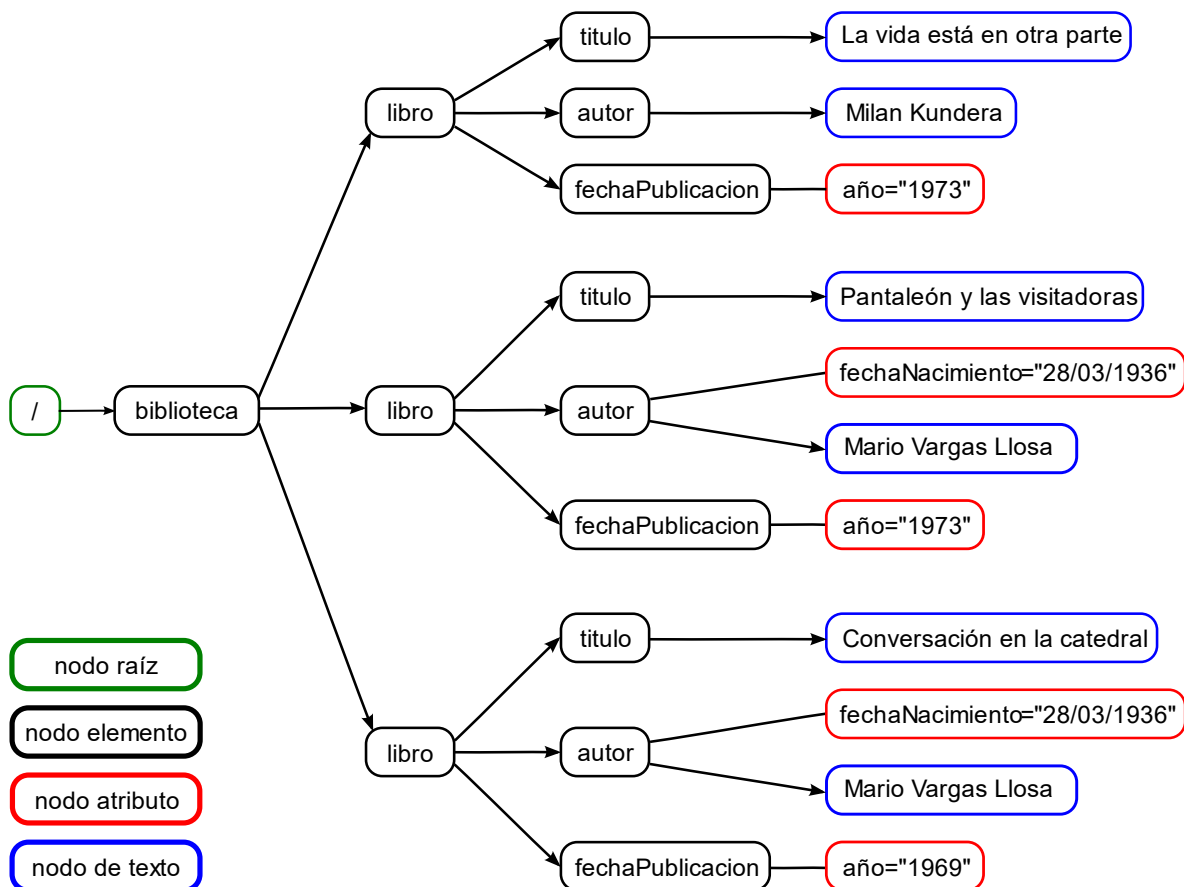
```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
</biblioteca>
```

```

</libro>
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>
</biblioteca>

```

se puede representar mediante el siguiente grafo:



Los nodos atributos y de texto no son como los nodos elemento. Por ejemplo, los nodos atributo y de texto **no pueden tener descendientes**. En realidad, **el nodo atributo ni siquiera se considera como hijo**, sino como una etiqueta adosada al elemento. **El texto** contenido por una etiqueta **sí que se considera hijo del elemento**, aunque las expresiones XPath suelen trabajar con nodos elemento y **para referirse a los atributos o al texto se utilizan notaciones especiales**.

1.3 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 el explorador de Windows o en la shell de GNU/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 escribir de dos formas distintas:

- **sintaxis abreviada:** más compacta y fácil de leer, que se explica en esta lección
- **sintaxis completa:** más larga, pero con más opciones disponibles

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, el texto que contienen o ambos.

1.4 Sintaxis abreviada

Veamos unos ejemplos de expresiones XPath de sintaxis abreviada y el resultado de su evaluación en el documento de ejemplo anterior:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

1.4.1 Eje

El eje nos permite seleccionar un subconjunto de nodos del documento y corresponde a recorridos en el árbol del documento. **Los nodos elemento se**

indican mediante el nombre del elemento. Los nodos atributo se indican mediante @ y el nombre del atributo.

- /: si está al principio de la expresión, indica el nodo raíz, si no, indica "hijo". Debe ir seguida del nombre de un elemento.

Ejemplos:

```
/biblioteca/libro/autor
```

```
<autor>Milan Kundera</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
/autor
```

No devuelve nada porque <autor> no es hijo del nodo raíz.

```
/biblioteca/autor
```

No devuelve nada porque <autor> no es hijo de <biblioteca>.

```
/biblioteca/libro/autor/@fechaNacimiento
```

```
fechaNacimiento="28/03/1936"  
fechaNacimiento="28/03/1936"
```

- //: indica "descendiente" (hijos, hijos de hijos, etc.).

```
/biblioteca//autor
```

```
<autor>Milan Kundera</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//autor
```

```
<autor>Milan Kundera</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//autor//libro
```

No devuelve nada porque <libro> no es descendiente de <autor>.

```
//@año
```

```
año="1973"  
año="1973"  
año="1969"
```

- /..: indica el elemento padre.

Nota: En el resultado de los ejemplos siguientes se obtienen únicamente los nodos que tienen el atributo fechaNacimiento.

```
/biblioteca/libro/autor/@fechaNacimiento/..
```

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//@fechaNacimiento/../../
```

```
<libro>  
  <titulo>Pantaleón y las visitadoras</titulo>  
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
  <fechaPublicacion año="1973"/>  
</libro>  
<libro>  
  <titulo>Conversación en la catedral</titulo>  
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
  <fechaPublicacion año="1969"/>  
</libro>
```

- |: permite indicar varios recorridos.

```
//autor|//titulo
```

```
<titulo>La vida está en otra parte</titulo>  
<autor>Milan Kundera</autor>  
<titulo>Pantaleón y las visitadoras</titulo>  
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```



```
<titulo>Conversación en la catedral</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//autor//titulo//@año
```

```
<titulo>La vida está en otra parte</titulo>
<autor>Milan Kundera</autor>
año="1973"
<titulo>Pantaleón y las visitadoras</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
año="1973"
<titulo>Conversación en la catedral</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
año="1969"
```

1.4.2 Predicado

El predicado se escribe entre corchetes, a continuación del eje. Si el eje ha seleccionado unos nodos, el predicado permite restringir esa selección a los que cumplan determinadas condiciones.

- **[@atributo]**: selecciona los elementos que tienen el atributo.

```
//autor[@fechaNacimiento]
```

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

- **[número]**: si hay varios resultados selecciona uno de ellos por número de orden; **last()** selecciona el último de ellos

```
//libro[1]
```

```
<libro>
  <titulo>La vida está en otra parte</titulo>
  <autor>Milan Kundera</autor>
  <fechaPublicacion año="1973"/>
</libro>
```

```
//libro[last()]
```

```
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>
```

```
//libro[last()-1]
```

```
<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>
```

- **[condicion]:** selecciona los nodos que cumplen la condición.

Los predicados permiten definir condiciones sobre los valores de los atributos. En las condiciones se pueden utilizar los operadores siguientes:

- operadores lógicos: and, or, not()
- operadores aritméticos: +, -, *, div, mod
- operadores de comparación: =, !=, <, >, <=, >=

Las comparaciones se pueden hacer entre valores de nodos y atributos o con cadenas de texto o numéricas. Las cadenas de texto deben escribirse entre comillas simples o dobles. En el caso de las cadenas numéricas, las comillas son optativas.

La condición puede utilizar el valor de un atributo (utilizando @) o el texto que contiene el elemento.

En los ejemplos siguientes se obtienen respectivamente los elementos <fechaPublicacion> cuyo atributo año es posterior/mayor a 1970 y los elementos <libro> cuyo subelemento <autor> tiene como contenido "Mario Vargas Llosa":

```
//fechaPublicacion[@año>1970]
```

Resultado:

```
<fechaPublicacion año="1973"/>
<fechaPublicacion año="1973"/>
```

```
//libro[autor="Mario Vargas Llosa"]
```

Resultado:

```
<libro>
```

```

<titulo>Pantaleón y las visitadoras</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1973"/>
</libro>
<libro>
<titulo>Conversación en la catedral</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1969"/>
</libro>

```

- Para hacer referencia al propio valor del elemento seleccionado se utiliza el punto (.).

```
//@año[.>1970]
```

Resultado:

```

año="1973"
año="1973"

```

```
//autor[.="Mario Vargas Llosa"]
```

Resultado:

```

<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>

```

- Un predicado puede contener **condiciones compuestas**.

En los ejemplos siguientes se seleccionan, respectivamente, los libros escritos por Mario Vargas Llosa **y** publicados en 1973 (primer ejemplo) y los libros escritos por Mario Vargas Llosa **o** publicados en 1973 (segundo ejemplo):

```
//libro[autor="Mario Vargas Llosa" and fechaPublicacion/@año="1973"]
```

Resultado:

```

<libro>
<titulo>Pantaleón y las visitadoras</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1973"/>
</libro>

```

```
//libro[autor="Mario Vargas Llosa" or fechaPublicacion/@año="1973"]
```

Resultado:

```
<libro>
```

```

<titulo>La vida está en otra parte</titulo>
<autor>Milan Kundera</autor>
<fechaPublicacion año="1973"/>
</libro>
<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>

```

- Se pueden escribir **varios predicados seguidos**, cada uno de los cuales restringe los resultados del anterior, **como si estuvieran encadenados por la operación lógica and**.

En el ejemplo siguiente se seleccionan los libros escritos por Mario Vargas Llosa y publicados en 1973:

```
//libro[autor="Mario Vargas Llosa"][fechaPublicacion/@año="1973"]
```

Resultado:

```

<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>

```

1.4.3 Selección de nodos

La selección de nodos **se escribe a continuación del eje y el predicado**. Si el eje y el predicado han seleccionado unos nodos, **la selección de nodos indica con qué parte de esos nodos nos quedamos**.

- **/node()**: selecciona todos los hijos (elementos o texto) del nodo.
- **//node()**: selecciona todos los descendientes (elementos o texto) del nodo.

```
//libro/node()
```

Resultado:

```

<titulo>La vida está en otra parte</titulo>
<autor>Milan Kundera</autor>
<fechaPublicacion año="1973"/>
<titulo>Pantaleón y las visitadoras</titulo>

```

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1973"/>
<titulo>Conversación en la catedral</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1969"/>
```

```
//autor/node()
```

Resultado:

```
Milan Kundera
Mario Vargas Llosa
Mario Vargas Llosa
```

```
//libro//node()
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>
La vida está en otra parte
<autor>Milan Kundera</autor>
Milan Kundera
<fechaPublicacion año="1973"/>
<titulo>Pantaleón y las visitadoras</titulo>
Pantaleón y las visitadoras
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
Mario Vargas Llosa
<fechaPublicacion año="1973"/>
<titulo>Conversación en la catedral</titulo>
Conversación en la catedral
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
Mario Vargas Llosa
<fechaPublicacion año="1969"/>
```

- **/text()**: selecciona únicamente el texto contenido en el nodo.
- **//text()**: selecciona únicamente el texto contenido en el nodo y todos sus descendientes.

```
//autor/text()
```

Resultado:

```
Milan Kundera
Mario Vargas Llosa
Mario Vargas Llosa
```

```
//libro/text()
```

Resultado:

No devuelve nada porque <libro> no contiene texto.

```
//libro//text()
```

Resultado:

```
La vida está en otra parte
Milan Kundera
Pantaleón y las visitadoras
Mario Vargas Llosa
Conversación en la catedral
Mario Vargas Llosa
```

- `/`: selecciona todos los hijos (sólo elementos) del nodo.
- `//`: selecciona todos los descendientes (sólo elementos) del nodo.

```
/biblioteca/*
```

Resultado:

```
<libro>
  <titulo>La vida está en otra parte</titulo>
  <autor>Milan Kundera</autor>
  <fechaPublicacion año="1973"/>
</libro>
<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>
```

```
//autor/*
```

Resultado:

No devuelve nada porque <autor> sólo contiene texto.

```
/biblioteca//*
```

Resultado:

```

<libro>
  <titulo>La vida está en otra parte</titulo>
  <autor>Milan Kundera</autor>
  <fechaPublicacion año="1973"/>
</libro>
<titulo>La vida está en otra parte</titulo>
<autor>Milan Kundera</autor>
<fechaPublicacion año="1973"/>
<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>
<titulo>Pantaleón y las visitadoras</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1973"/>
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>
<titulo>Conversación en la catedral</titulo>
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
<fechaPublicacion año="1969"/>

```

- /@*: selecciona todos los atributos del nodo.
- //@*: selecciona todos los atributos de los descendientes del nodo.

```
//@*
```

Resultado:

```

año="1973"
fechaNacimiento="28/03/1936"
año="1973"
fechaNacimiento="28/03/1936"
año="1969"

```

```
//libro/@*
```

Resultado:

No devuelve nada porque <libro> no tiene atributos

```
//autor/@*
```

Resultado:

```
fechaNacimiento="28/03/1936"  
fechaNacimiento="28/03/1936"
```

1.5 Pasos de búsqueda consecutivos

Una expresión XPath puede contener varios pasos de búsqueda consecutivos. Cada uno incluirá su eje (y en su caso, su predicado) y el último paso de búsqueda incluirá en su caso una selección de nodos. Cada paso de búsqueda trabaja a partir de los nodos seleccionados por el paso de búsqueda anterior.

En el ejemplo siguiente se obtienen los títulos de los libros publicados después de 1970, mediante dos pasos de búsqueda:

- en el primer paso (`//fechaPublicacion[@año>1970]`) se seleccionan los elementos `<fechaPublicacion>` cuyo atributo `año` es superior a 1970.
- en el segundo paso (`/../titulo`), se seleccionan primero los elementos padre (`/..`) de los `<fechaPublicacion>` seleccionados en el primer paso de búsqueda (es decir, elementos `<libro>`) y a continuación sus subelementos `<titulo>`.

```
//fechaPublicacion[@año>1970]/../titulo
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>  
<titulo>Pantaleón y las visitadoras</titulo>
```

Un determinado resultado se puede obtener mediante un sólo paso de búsqueda o mediante varios pasos.

En los ejemplos siguientes se obtienen los libros escritos por Mario Vargas Llosa de dos formas distintas:

- Mediante un sólo paso de búsqueda. Se seleccionan los elementos `<libro>` cuyo subelemento `<autor>` tiene como contenido la cadena "Mario Vargas Llosa".

```
//libro[autor="Mario Vargas Llosa"]
```

Resultado:

```
<libro>  
  <titulo>Pantaleón y las visitadoras</titulo>  
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
  <fechaPublicacion año="1973"/>  
</libro>  
<libro>  
  <titulo>Conversación en la catedral</titulo>  
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
  <fechaPublicacion año="1969"/>
```



```
</libro>
```

- mediante dos pasos de búsqueda. En el primer paso se seleccionan los elementos <autor> cuyo contenido es la cadena "Mario Vargas Llosa". En el segundo paso de búsqueda se seleccionan los elementos padre (es decir, los elementos <libro>).

```
//autor[.="Mario Vargas Llosa"]/..
```

Resultado:

```
<libro>
  <titulo>Pantaleón y las visitadoras</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1973"/>
</libro>
<libro>
  <titulo>Conversación en la catedral</titulo>
  <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
  <fechaPublicacion año="1969"/>
</libro>
```

En los ejemplos siguientes se obtiene el autor que ha publicado libros en 1969 de varias formas distintas:

```
//@año[.="1969"]/../../autor
```

Resultado:

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//libro[fechaPublicacion/@año=1969]/autor
```

Resultado:

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//fechaPublicacion[@año=1969]../../autor
```

Resultado:

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

```
//autor[../fechaPublicacion/@año=1969]
```

Resultado:

```
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
```

1.6 Expresiones anidadas

Las expresiones XPath pueden anidarse, lo que permite definir expresiones más complicadas. Por ejemplo, en el documento utilizado anteriormente:

Un ejemplo de expresión anidada sería, por ejemplo, obtener los títulos de los libros publicados el mismo año que la novela "La vida está en otra parte". Esta información no está directamente almacenada en el documento, pero se puede obtener la respuesta en dos pasos:

1. obtener primero el año en que se publicó la novela "La vida está en otra parte":

```
//libro[titulo="La vida está en otra parte"]/fechaPublicacion/@año
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>  
<titulo>Pantaleón y las visitadoras</titulo>
```

2. y obtener después los títulos de los libros publicados en 1973:

```
//libro[fechaPublicacion/@año=1973]/titulo
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>  
<titulo>Pantaleón y las visitadoras</titulo>
```

Estas dos expresiones se pueden unir en una única expresión, sustituyendo en la segunda expresión el valor 1973 por la primera expresión:

```
//libro[fechaPublicacion/@año=//libro[titulo="La vida está en otra parte"]/fechaPublicacion/@año]/titulo
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>  
<titulo>Pantaleón y las visitadoras</titulo>
```

Como cada una de las expresiones puede escribirse de varias maneras, en realidad hay muchas formas de encontrar la respuesta. Por ejemplo, en la solución siguiente los predicados se encuentran al final del eje en cada subexpresión:

```
//titulo[../fechaPublicacion/@año=//@año[../titulo="La vida está en otra parte"]]
```

Resultado:

```
<titulo>La vida está en otra parte</titulo>  
<titulo>Pantaleón y las visitadoras</titulo>
```

Otro ejemplo de expresión anidada sería obtener los títulos de los libros del mismo autor que la novela "Pantaleón y las visitadoras". Como en el ejemplo anterior, la respuesta puede obtenerse en dos pasos:

1. Obtener primero el autor de la novela "Pantaleón y las visitadoras":

```
//libro[titulo="Pantaleón y las visitadoras"]/autor/text()
```

Resultado:

```
Mario Vargas Llosa
```

2. Y obtener después los títulos de los libros escritos por Mario Vargas Llosa:

```
//libro[autor="Mario Vargas Llosa"]/titulo
```

Resultado:

```
<titulo>Pantaleón y las visitadoras</titulo>
<titulo>Conversación en la catedral</titulo>
```

Estas dos expresiones se pueden unir en una única expresión, sustituyendo en la segunda expresión el valor "Mario Vargas Llosa" por la primera expresión:

```
//libro[autor=//libro[titulo="Pantaleón y las visitadoras"]/autor/text()]/titulo
```

Resultado:

```
<titulo>Pantaleón y las visitadoras</titulo>
<titulo>Conversación en la catedral</titulo>
```

Un detalle importante es que no hay que escribir la primera expresión entre comillas.

Incluso **se puede omitir la selección de nodos /text()** de la segunda expresión y escribir la expresión XPath así:

```
//libro[autor=//libro[titulo="Pantaleón y las visitadoras"]/autor]/titulo
```

Resultado:

```
<titulo>Pantaleón y las visitadoras</titulo>
<titulo>Conversación en la catedral</titulo>
```

2. XSL: Lenguaje de Hojas de Estilo Extensible

Aunque las hojas de estilo **CSS** se pueden aplicar a los documentos XML, las hojas de estilo **tienen muchas limitaciones** ya que se crearon para complementar al HTML, es decir, para ver páginas web en pantallas de ordenador. De la misma manera que el **XML es una generalización del HTML**, el **W3C creó una generalización de las hojas de estilo CSS a la que se denominó XSL**

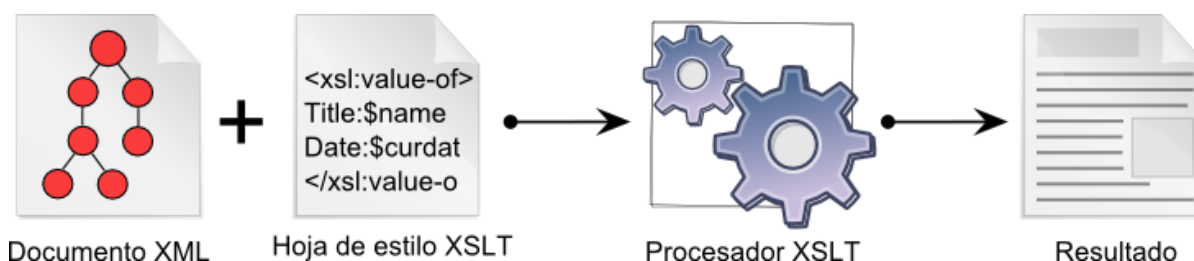
(eXtensible Stylesheet Language), es decir, Lenguaje de hojas de estilo extensible.

El W3C ha desarrollado tres lenguajes:

- **XPath**: un lenguaje para referirse a partes de un documento XML.
- **XSLT** (XSL Transformation, es decir, Transformaciones XSL): un lenguaje para transformar documentos XML.
- XSL-FO (XSL Formatting Objects, es decir, Objetos de formato XSL): un lenguaje para especificar el formato de un documento XML y posteriormente convertirlo a PDF o PostScript.

2.1 XSLT

XSLT (Transformaciones XSL) es un lenguaje de programación declarativo que permite generar documentos a partir de documentos XML, como ilustra la imagen siguiente:



- El documento XML es el documento inicial a partir del cual se va a generar el resultado.
- La hoja de estilo XSLT es el documento que contiene el código fuente del programa, es decir, las reglas de transformación que se van a aplicar al documento inicial.
- El procesador XSLT es el programa de ordenador que aplica al documento inicial las reglas de transformación incluidas en la hoja de estilo XSLT y genera el documento final.
- El resultado de la ejecución del programa es un nuevo documento (que puede ser un documento XML o no).

XSLT se utiliza para obtener a partir de un documento XML otros documentos (XML o no). A un documento XML se le pueden aplicar distintas hojas de estilo XSLT para obtener distintos resultados y una misma hoja de estilo XSLT se puede aplicar a distintos documentos XML.

3. Referencias:

<https://www.mclibre.org/consultar/xml/lecciones/xml-xpath.html#>