



# Qué es XPath

Path: Ruta, camino...

#### Definición:

El lenguaje Xpath es el sistema que se utiliza para navegar y consultar los elementos y atributos contenidos en la estructura de un documento XML.

#### Para qué sirve:

XPath sirve para que los programadores puedan definir criterios de búsqueda avanzada y cálculos específicos, utilizando una sintaxis simple, pero bastante eficaz.

#### **Aplicaciones:**

XML Copy Editor





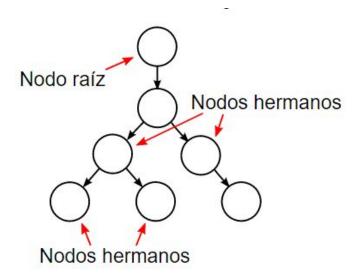
# Árbol del documento

XPath considera un documento XML como un árbol de nodos:

Un **árbol** es un grafo en el que cualquier pareja de vértices están conectada por un único camino (es decir, que no hay ciclos). Un **árbol dirigido** es un árbol en el que las aristas tienen dirección y todos los nodos menos uno tienen un único padre.

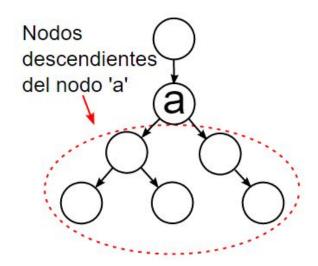
## Nodo raíz

El **nodo raíz** de un árbol dirigido es el único **nodo sin padre**. Los **nodos hermanos** son los nodos que tienen el **mismo padre**.



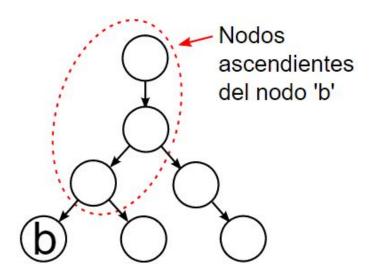
## Nodos descendientes

Los nodos **descendientes** de un nodo son todos los nodos a los que se llega desde el nodo: los hijos, los hijos de los hijos, etc.



## Nodos ascendientes

Los nodos **ascendientes** de un nodo son todos los nodos de los que un nodo es descendiente: el padre, el padre del padre, etc.



```
Nodo raíz: <bookstore>

Nodos hermanos: <title> <author> <year> <price>

Nodos descendientes del elemento <book>: <title> <author> <year> <price>

Nodos ascendientes del elemento <price>: <book> <bookstore>
```

# Tipos de nodos

# 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

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

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.

```
Nombre
                                                                                                             Ana
<empresa>
     <empleado sexo="mujer">
                                                                                               Apellidos
                                                                                                            López
                                                                                  Empleado
          <nombre>Ana</nombre>
          <apellido>Lopez</apellido>
                                                                                                             Calle
                                                                                                                          Leonardo
          <direccion>
                                                                                              Dirección
               <calle>Leonardo</calle>
               <numero>20</numero>
                                                                                                             Número
               <ciudad cp="39200">Santander</ciudad</pre>
          </direccion>
                                                                                                                         CP="39200"
                                                                                                             Ciudad
                                                                        Empresa
     </empleado>
                                                                                                                         Santander
     <empleado sexo="hombre">
          <nombre>Luis</nombre>
                                                                                              ► Sexo="hombre
          <apellido>Ríos</apellido>
          <direccion>
               <calle>General</calle>
                                                                                                 Nombre
                                                                                                               Luis
               <numero>3</numero>
               <ciudad cp="39120">Torrelavega</ciu</pre>
                                                                                                 Apellidos
                                                                                                               Ríos
          </direccion>
                                                                                     Empleado
     </empleado>
                                                                                                               Calle
                                                                                                                            General
                                                                                                Dirección
</empresa>
                                                                                                               Número
                                                                                                                           CP="39120"
                                                                                                               Ciudad
                                                                                                                           Torrelavega
```

Sexo="muier

# Sintaxis de la 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.
- 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.

## 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
   <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 catedrak/titulo>
   <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa/autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

# Eje (en inglés, axis)

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.

/biblioteca/libro/autor	<pre><autor>Milan Kundera </autor>   <autor fechanacimiento="28/03/1936">Mario Vargas Llosa </autor>   <autor fechanacimiento="28/03/1936">Mario Vargas Llosa </autor></pre> Vargas Llosa
/autor	No devuelve nada porque <autor> no es hijo del nodo raíz.</autor>
/biblioteca/libro/autor/@fechaNacimiento	<pre>fechaNacimiento = "28/03/1936" fechaNacimiento = "28/03/1936"</pre>

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

//autor	<autor>Milan Kundera</autor> <autor fechanacimiento="28/03/1936">Mario Vargas Llosa</autor> <autor fechanacimiento="28/03/1936">Mario Vargas Llosa</autor>
//@año	año="1973" año="1973" año="1969"

#### /..: indica el elemento padre.

<autor fechaNacimiento = "28/03/1936">Mario /biblioteca/libro/autor/@fechaNacimiento/... Vargas Llosa</autor> //@fechaNacimiento/.. <autor fechaNacimiento = "28/03/1936" > Mario Vargas Llosa</autor> libro> //@fechaNacimiento/../.. <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.

//titulo //@año	<titulo>La vida está en otra parte</titulo> año="1973" <titulo>Pantaleón y las visitadoras</titulo> año="1973" <titulo>Conversación en la catedral</titulo> año="1969"	•
//autor //titulo	<pre><titulo>La vida está en otra parte </titulo></pre>	
		22

## 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]	<li>libro&gt;</li>	
	<titulo>La vida está en otra parte </titulo> <autor>Milan Kundera </autor> <fechapublicacion año="1973"></fechapublicacion>	
//libro[last()]	<pre><li><li><li>titulo&gt;Conversación en la catedral          <autor fechanacimiento="28/03/1936">Mario Vargas Llosa </autor></li></li></li></pre>	
//libro[last()-1]	<pre>libro[last()-1]  <li><li><li>titulo&gt;Pantaleón y las visitadoras </li></li></li></pre>	

[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.

```
//fechaPublicacion [@año>1970]
                                      <fechaPublicacion año="1973"/>
                                      <fechaPublicacion año="1973"/>
//libro[autor="Mario Vargas Llosa"]
                                      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>
```

Un predicado puede contener condiciones compuestas.

Los libros escritos por Mario Vargas Llosa y publicados en 1973

#### y los libros escritos por Mario Vargas Llosa o publicados en 1973:

```
//libro[autor="Mario Vargas Llosa" or
                                        libro>
fechaPublicacion/@año="1973"1
                                          <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:

# Ejemplo1

Título de los libros publicados en 1973

Autor de los libros publicados antes del 73

El primer libro escrito por Mario Vargas Llosa

Autor del libro 'La vida está en otra parte'

# Ejemplo2

```
<?xml version="1.0" encoding="UTF-8"?>
<floristeria>
 <flor>
  <nombre>Clavel</nombre>
  cio>23€</precio>
  <temporada>Primavera</temporada>
  <origen nLote="213">España</origen>
 </flor>
 <flor>
  <nombre>Rosa</nombre>
  cio>23€</precio>
  <temporada>Verano</temporada>
  <origen nLote="225">España</origen>
 </flor>
  <nombre>Margarita</nombre>
_ cio>30€</precio>
<temporada>Primavera</temporada>
```

Nombre de las flores que cuestan 23€

Precio de las flores con número lote mayor que 200

Todas las flores que sean de España o Italia

Número de lote de las flores de España con precio 30€

Nombre de la anteúltima flor

# Selección de nodos (node test)

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.

<pre><titulo>La vida está en otra parte </titulo></pre>	
Milan Kundera Mario Vargas Llosa Mario Vargas Llosa	
<pre><titulo>La vida está en otra parte </titulo> La vida está en otra parte <autor>Milan Kundera </autor> Milan Kundera <fechapublicacion año="1973"></fechapublicacion> <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"></fechapublicacion> <titulo>Conversación en la catedral </titulo> Conversación en la catedral <autor fechanacimiento="28/03/1936">Mario Vargas Llosa </autor> Mario Vargas Llosa</pre>	

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

//autor/text()	Milan Kundera Mario Vargas Llosa Mario Vargas Llosa
//libro/text()	No devuelve nada porque <libro> no contiene texto.</libro>
//libro//text()	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/
              ibro>
                <titulo>La vida está en otra parte </titulo>
                <autor>Milan Kundera
                <fechaPublicacion año="1973"/>
              </libro>
              ibro>
                <titulo>Pantaleón y las visitadoras </titulo>
                <autor fechaNacimiento = "28/03/1936">Mario Vargas Llosa </autor>
                <fechaPublicacion año="1973"/>
              </libro>
              ibro>
                <titulo>Conversación en la catedral </titulo>
                <autor fechaNacimiento = "28/03/1936">Mario Vargas Llosa </autor>
                <fechaPublicacion año="1969"/>
              </libro>
//autor/*
              No devuelve nada porque <autor> sólo contiene texto.
/biblioteca/
              libro>
                <titulo>La vida está en otra parte </titulo>
                <autor>Milan Kundera
                <fechaPublicacion año="1973"/>
                                                                                              35
              </libro>
              <titulo>La vida está en otra parte </titulo>
```

/@\*: selecciona todos los atributos del nodo.

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

//@*	año="1973" fechaNacimiento="28/03/1936" año="1973" fechaNacimiento="28/03/1936" año="1969
//libro/@*	No devuelve nada porque <libro> no tiene atributos</libro>
//autor/@*	fechaNacimiento = "28/03/1936" fechaNacimiento = "28/03/1936"

**Nota**: En XPath 1.0 no se puede seleccionar únicamente el valor del atributo, sino que se obtienen respuestas del tipo nombreDelAtributo=ValorDelAtributo

## Ejercicio 1

Dado el siguiente documento XML, escriba las expresiones XPath que devuelvan la respuesta deseada (mostrada en los cuadros).

```
<?xml version="1.0" encoding="UTF-8"?>
<ies>
  <nombre>IES Linares</nombre>
  <web>http://www.aglinares.com</web>
  <ciclos>
   <ciclo id="ASTR">
     <nombre>Administración de Sistemas Informáticos en Red
     <grado>Superior
     <decretoTitulo año="2009" />
   </ciclo>
   <ciclo id="DAW">
     <nombre>Desarrollo de Aplicaciones Web</nombre>
     <grado>Superior
     <decretoTitulo año="2010" />
   </ciclo>
   <ciclo id="SMR">
     <nombre>Sistemas Microinformáticos y Redes</nombre>
     <grado>Medio</grado>
     <decretoTitulo año="2008" />
   </ciclo>
  </ciclos>
```

#### XPath - Ejercicio 1-1

#### Nombre del Instituto:

<nombre>IES Linares

## XPath - Ejercicio 1-2

#### Página web del Instituto:

http://www.aglinares.com

## XPath - Ejercicio 1-3

#### Nombre de los Ciclos Formativos:

Administración de Sistemas Informáticos en Red Desarrollo de Aplicaciones Web Sistemas Microinformáticos y Redes

## XPath - Ejercicio 1-4

Siglas por las que se conocen los Ciclos Formativos:

```
id="ASIR"
id="DAW"
id="SMR"
```

## XPath - Ejercicio 1-5

Años en los que se publicaron los decretos de título de los Ciclos Formativos:

```
año="2009"
año="2010"
año="2008"
```

## XPath - Ejercicio 1-6

Ciclos Formativos de Grado Medio (se trata de obtener el elemento <ciclo> completo):

```
<ciclo id="SMR">
  <nombre>Sistemas Microinformáticos y Redes</nombre>
  <grado>Medio</grado>
  <decretoTitulo año="2008"/>
</ciclo>
```

#### XPath - Ejercicio 1-7

#### Nombre de los Ciclos Formativos de Grado Superior:

<nombre>Administración de Sistemas Informáticos en Red<nombre>Desarrollo de Aplicaciones Web

## XPath - Ejercicio 1-8

#### Nombre de los Ciclos Formativos anteriores a 2010:

Administración de Sistemas Informáticos en Red Sistemas Microinformáticos y Redes

## XPath - Ejercicio 1-9

Nombre de los Ciclos Formativos de 2008 o 2010:

Desarrollo de Aplicaciones Web Sistemas Microinformáticos y Redes

## Ejercicio 2

Dado el siguiente documento XML, escriba las expresiones XPath que devuelvan la respuesta deseada (mostrada en los cuadros).

```
<?xml version="1.0" encoding="UTF-8"?>
<ies>
  <modulos>
    <modulo id="0228">
      <nombre>Aplicaciones web </nombre>
      <curso>2</curso>
      <horasSemanales > 4</horasSemanales >
      <ciclo>SMR</ciclo>
    </modulo>
    <modulo id="0372">
      <nombre>Gestión de bases de datos </nombre>
      <curso>1</curso>
      <horasSemanales >5</horasSemanales >
      <ciclo>ASIR</ciclo>
    </module>
    <modulo id="0373">
      <nombre>Lenguajes de marcas y sistemas de gestión de información </nombre>
      <curso>1</curso>
      <horasSemanales >3</horasSemanales >
      <ciclo>ASIR</ciclo>
      <ciclo>DAW</ciclo>
    </modulo>
    <modulo id="0376">
      <nombre>Implantación de aplicaciones web </nombre>
      <curso>2</curso>
      <horasSemanales > 5 /horasSemanales >
      <ciclo>ASIR</ciclo>
    </modulo>
  </modulos>
```

#### XPath - Ejercicio 2-1

Nombre de los módulos que se imparten en el Instituto:

Aplicaciones web Gestión de bases de datos Lenguajes de marcas y sistemas de gestión de información Implantación de aplicaciones web

## XPath - Ejercicio 2-2

Nombre de los módulos del ciclo ASIR:

Gestión de bases de datos Lenguajes de marcas y sistemas de gestión de información Implantación de aplicaciones web

## XPath - Ejercicio 2-3

Nombre de los módulos que se imparten en el segundo curso de cualquier ciclo:

Aplicaciones web Implantación de aplicaciones web

## XPath - Ejercicio 2-4

Nombre de los módulos de menos de 5 horas semanales:

```
Aplicaciones web
Lenguajes de marcas y sistemas de gestión de información
```

## XPath - Ejercicio 2-5

Nombre de los módulos que se imparten en el primer curso de ASIR:

```
Gestión de bases de datos
Lenguajes de marcas y sistemas de gestión de información
```

## XPath - Ejercicio 2-6

Horas semanales de los módulos de más de 3 horas semanales:

```
455
```

## Expresiones anidadas

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

```
<?xml version="1.0" encoding="UTF-8"?>
<br/>biblioteca>
  libro>
   <titulo>La vida está en otra parte/titulo>
   <autor>Milan Kundera
   <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 catedrak/titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa/autor>
   <fechaPublicacion año="1969"/>
_</libro>
</biblioteca>
```

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:

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

Solución: año="1973"

• y obtener después los títulos de los libros publicados en 1973:

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

Solución:

<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
```

#### Solución:

- <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.

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:

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

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

Obtener después los títulos de los libros de "Mario Vargas Llosa"

//libro[autor="Mario Vargas Llosa"]/titulo/text()

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/text()

Nota: Debes quitar las "" al sustituir "Mario Vargas Llosa" por una expresión. Podemos quitar el text() también:

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

## Ejercicio 3: Expresiones anidadas

Dada la base de datos de una protectora, escriba las siguientes expresiones anidadas XPath que devuelvan la respuesta deseada (mostrada en los cuadros):

- El nombre del perro del dueño 'D02'.
- Nombre del dueño de Visi.
- Ciudades donde viven los Fox Terrier.
- Tamaño de los perros que no viven en Santander.

xml version="1.0" encoding="UTF-8"?	
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	<dueños></dueños>
<pre><perros></perros></pre>	
<perro id="P01"></perro>	<dueño id="D01"></dueño>
<nombre>Yogui</nombre>	<nombre>Paz</nombre>
<raza>Fox Terrier</raza>	<apellido>Hernandez</apellido>
<tamaño>pequeño</tamaño>	<ciudad>Santander</ciudad>
<peso>8</peso>	<n_perros>2</n_perros>
	<id perro="">P01</id>
	<id_perro>P02</id_perro>
<perro id="P02"></perro>	
<nombre>Visi</nombre>	
<raza>Fox Terrier</raza>	<dueño id="D02"></dueño>
<tamaño>pequeño</tamaño>	<nombre>Noa</nombre>
<peso>5</peso>	<apellido>García</apellido>
	<ciudad>Torrelavega</ciudad>
	<n_perros>1</n_perros>
<perro id="P03"></perro>	<id_perro>P03</id_perro>
<nombre>Marla</nombre>	
<raza>American Stanford</raza>	
<tamaño>mediano</tamaño>	<dueño id="D03"></dueño>
<peso>13</peso>	<nombre>Pedro</nombre>
	<apellido>Ríos</apellido>
	<ciudad>Astillero</ciudad>
<pre><perro id="P04"></perro></pre>	<n_perros>2</n_perros>
<nombre>Laika</nombre>	<id_perro>P04</id_perro>
<raza>Fox Terrier</raza>	<id_perro>P05</id_perro>
<tamaño>pequeño</tamaño>	
<pre><peso>7</peso></pre>	
<pre><perro id="P05"></perro></pre>	
<nombre>Bepo</nombre>	
<raza>schnauzer</raza>	
<tamaño>grande</tamaño>	
<pre><peso>28</peso></pre>	

Solución: El nombre del perro del dueño 'D02': El identificador del perro del dueño D02: //dueño[@id="D02"]/id\_perro el nombre del perro con identificador:P03 //perro[@id="P03"]/nombre //perro[@id=//dueño[@id="D02"]/id\_perro]/nombre Nombre del dueño de Visi: identificador de Visi: //perro[nombre="Visi"]/@id Nombre del dueño de id\_prro=P02: //dueño[id\_perro="P02"]/nombre //dueño[id\_perro=//perro[nombre="Visi"]/@id]/nombre Ciudades donde viven los Fox Terrier: Id de los fox terrier: //perro[raza="Fox Terrier"]/@id ciudad donde vive el dueño de los perros con id0X: //dueño[id\_perro="P01"]/ciudad //dueño[id\_perro=//perro[raza="Fox Terrier"]/@id]/ciudad Tamaño de los perros que no viven en Santander: id de los perros que no viven en Santander:

//dueño[ciudad!="Santander"]/id\_perro

Tamaño de los perros con id "P03": //perro[@id="P03"]/tamaño

//perro[@id=//dueño[ciudad!="Santander"]/id\_perro]/tamaño

## **Funciones Xpath**

#### count(nodo):

Nos cuenta las veces que se repite ese nodo.

- Ej:Calcular el número de perros que tenemos en nuestra base de datos:
   count(//perro)
- Ej: Calcular el número de perros de tamaño pequeño: count(//perro[tamaño="pequeño"])

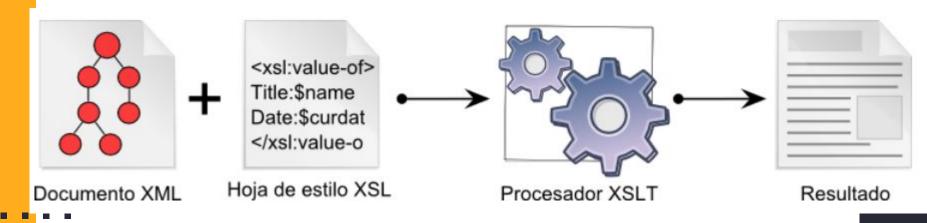
### Sum(nodo)

Suma el peso de todos los perros: sum(//peso)

# Conversión y adaptación de documentos(XSL)

## **XSLT: Transformaciones XSL**

XSLT (Transformaciones XSL) es un lenguaje de programación declarativo que permite generar documentos a partir de documentos XML:



- El documento XML es el documento inicial a partir del cual se va a generar el resultado.
- La **hoja de estilo XSL** 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 XSL 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,html,Xhtml).

A un documento XML se le pueden aplicar distintas hojas de estilo XSL para obtener distintos resultados y una misma hoja de estilo XSL se puede aplicar a distintos documentos XML.

## ¿Qué es XSLT?

Las hojas de estilo XSL no se escriben como una secuencia de instrucciones, sino como una colección de plantillas (template rules). Cada plantilla establece cómo se transforma un determinado elemento (definido mediante expresiones XPath). La transformación del documento se realiza de la siguiente manera:

- El procesador analiza el documento y construye el árbol del documento.
- El procesador recorre el árbol del documento desde el nodo raíz.
- En cada nodo recorrido, el procesador aplica o no alguna plantilla:
- Si a un nodo no se le puede aplicar ninguna plantilla, su contenido se incluye en el documento final (el texto del nodo, no el de los nodos descendientes). A continuación, el procesador recorre sus nodos hijos.
- Si a un nodo se le puede aplicar una plantilla, se aplica la plantilla. La plantilla puede generar texto que se incluye en el documento final. En principio, el procesador no recorre sus nodos hijos, salvo que la plantilla indique al procesador que sí que deben recorrerse los nodos hijos.
- Cuando el procesador ha recorrido el árbol, se ha terminado la transformación.

Una hoja de estilo XSL es un documento XML que contiene al menos las etiquetas siguientes:

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform " version="1.0">
</xsl:stylesheet>
```

## Estas etiquetas son:

- La declaración xml **<?xml>**, propia de cualquier documento XML.
- La instrucción **<xsl:stylesheet>** es la etiqueta raíz de la hoja de estilo, sus atributos indican la versión y el espacio de nombres correspondiente.

**Dentro de la instrucción <xsl:stylesheet>** se pueden encontrar los llamados elementos de alto nivel y las plantillas, como en el ejemplo siguiente:

Estas etiquetas son:

El elemento de alto nivel **<xsl:output>** indica el tipo de salida producida.

La instrucción <xsl:template> es una plantilla.

El atributo match indica los elementos afectados por la plantilla y contiene una expresión XPath.

Cuando se aplica una plantilla a un nodo, en principio no se recorren los nodos descendientes. Para indicar que **sí queremos recorrer los nodos descendientes** y **aplicarles las plantillas** que les correspondan, hay que utilizar la instrucción **<xsl:apply-templates />**, como en el ejemplo siguiente:

## Enlazar documentos XML con hojas de estilo XSL

Se puede asociar de forma permanente una hoja de estilo XSL a un documento XML mediante la instrucción de procesamiento <?xml-stylesheet ?>, la misma que permite asociar hojas de estilo CSS. La instrucción de procesamiento <?xml-stylesheet ... ?> va al principio del documento, después de la declaración XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="ejemplo.xsl"?>
```

Cuando se visualiza en un navegador web un documento XML enlazado con una hoja de estilo XSL, los navegadores muestran el resultado de la transformación, aunque si se muestra el código fuente de la página, los navegadores muestran el documento XML original.

**Nota:** Varios navegadores no muestra los documentos XML que enlazan a hojas de estilo XSL. Utilizar el internet explorer.

## Ejemplos de plantillas XSL

Vamos a ver ejemplos de plantillas trabajando sobre el documento siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="XSL Ejemplo1.xsl"?>
<catalogo>
    libro>
        <titulo>Manual imprescindible de C/C++</titulo>
        <isbn>9788441526143</isbn>
        <autores>
                <autor>Miguel Angel</autor>
        </autores>
        <paginas>416</paginas>
        <editorial>Anaya Multimedia</editorial>
    </libro>
    libro>
        <titulo>Redes locales</titulo>
        <isbn>9788441519800</isbn>
        <autores>
             <autor>Jim Doherty</autor>
             <autor>Neil Anderson</autor>
        </autores>
        <paginas>544</paginas>
        <editorial>Rama</editorial>
    </libro>
</catalogo>
```

## Crear la plantilla

En el caso que nos ocupa, queremos obtener un documento HTML, luego la plantilla deberá tener las siguientes líneas básicas, propias de cualquier documento HTML:

Entonces, el documento XSL con el que estábamos trabajando quedará así:

#### ESTRUCTURA BÁSICA XSL:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
 <xsl:output method="html" encoding="UTF-8" indent="yes"/>
 <xsl:template match="/">
   <html>
     <head>
       <title>Floristeria</title>
       <link rel="stylesheet" type="text/css" href="estilo.css"/>
     </head>
     <body>
                                                                               Documento html
       <div>
        <br></br>
        <h1>Lista Nombres</h1>
        </div>
     </body>
   </html>
 </xsl:template>
</xsl:stylesheet>
```

Como podemos ver la plantilla puede contener una página web completa, pero ¿Cómo indicamos en el documento XSL los datos del documento XML que queremos incluir?

## El elemento <xsl: value-of>

El elemento <xsl: value-of> se utiliza para extraer el valor de un nodo seleccionado.

#### **Elemento simple:**

#### xsl: value - of

Sintaxis: <xsl:value-of select="elementoXML" />

Para seleccionar el elemento XML debemos escribir la ruta Xpath.

Ejemplos: seleccionar nombres de las flores:

<xsl:value-of select="/floristeria/flor/nombre"/>

<xsl:value-of select="//nombre" />

## Elemento XSLT <xsl: for-each>

El elemento XSL <xsl: for-each> se puede utilizar para seleccionar cada elemento XML de un conjunto de nodos especificado:

```
<xsl:for-each select='*'>Elemento Compuesto<xsl:for-each>
```

También podemos filtrar la salida del archivo XML agregando un criterio al atributo de selección en el elemento <xsl: for-each>.

Por ejemplo: El precio de las flores de cuya temporada es primavera:

Nota: Cuidado en el uso " y ' .

# EJEMPLO <xsl: value-of> Tenemos el siguiente documento XML:

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="plantilla2.xsl"?>
<floristeria>
<flor>
        <nombre>Rosa</nombre>
 cio>10€</precio>
        <color>Rojo</color>
        <temporada>Primavera</temporada>
</flor>
<flor>
        <nombre>Narciso</nombre>
 cio>15€</precio>
        <color>Blanco</color>
        <temporada>Verano</temporada>
</flor>
<flor>
        <nombre>Margarita</nombre>
 cio>8€</precio>
        <color>Blanco</color>
        <temporada>Primavera</temporada>
</flor>
<flor>
        <nombre>Clavel</nombre>
 cio>25€</precio>
        <color>Rosa</color>
        <temporada>Primavera</temporada>
</flor>
</floristeria>
```

#### EJEMPLO <xsl: value-of>

Construye una plantilla XSL y transforma el documento floristeria.xml en un documento HTML que contenga la siguiente información:

- Lista con el nombre de cada flor.
- Construye un CSS y asígnale al documento html para que tenga una presentación armónica.

Plantilla XSL:	Hoja de estilo CSS
<pre><?xml version="1.0" encoding="UTF-8"?> <xsl:stylesheet version="1.0" xmlns="http://www.w3.org/1999/xhtml" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"></xsl:stylesheet></pre>	body{ font-family: Verdana; background: green; text-align: center;
<pre><html> <head>   <title>Floristeria</title>   <li>link rel="stylesheet" type="text/css" href="estilo1.css"/&gt;   </li></head>   <body>   <div>     <h1>Nombres</h1>   <ul>     <xsl:for-each select="floristeria/flor">       <li><xsl:value-of select="nombre"></xsl:value-of></li>     </xsl:for-each>   </ul> </div></body></html></pre>	h1{

## Ejemplo 0

Construye una plantilla XSL y transforma el documento EJ0.xml en un documento HTML que contenga la siguiente información:

- Lista con el cantante de cada disco.
- Tabla con el título del disco, nombre del cantante y el año de cada disco.

Construye un CSS y asígnale al documento html para que tenga una presentación armónica. Procura utilizar un elemento contenedor y trata de centrarlo horizontalmente.

<pre><?xml-stylesheet type="text/xsl" href="plantilla1.xsl"?> <catalogo></catalogo></pre>	
<titulo>Fiesta</titulo> <cantante>Manu Laso</cantante> <guitarra>Laura Lisa</guitarra> <bateria>Nano Meng</bateria> }	list-style-position: inside;  *Tabla**/ able {  background-color: lightgreen; border: 1px solid black; margin: auto;   {  border: 1px solid black; text-align: center; }

XSL

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
    <xsl:output method="html" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Catalogo</title>
        k rel="stylesheet" type="text/css" href="Ej0.css"/>
      </head>
      <body>
        <div>
          <hr></hr>
          <h1>Lista Cantantes</h1>
          <xsl:for-each select="catalogo/disco">
              <xsl:value-of select="cantante"/>
            </xsl:for-each>
          <h1>Tabla con el título, nombre del cantante y año de publicación</h1>
          TítuloNombreAño
           <xsl:for-each select="catalogo/disco">
           <xs:value-of select="titulo"/><xs:value-of select="cantante"/><xs:value-of select="año"/>
           </xsl:for-each>
          <br>></br>
        </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

# Ejemplo 1

Construye una plantilla XSL y transforma el documento EJ1.xml en un documento HTML que contenga la siguiente información:

- Lista con el título de cada libro y la editorial.
- Tabla con el título del libro, autores y editorial.

Construye un CSS y asígnale al documento html para que tenga una presentación armónica.

```
<?xml version="1.0" encoding="utf-8"?>
<catalogo>
libro>
  <titulo>Manual imprescindible de C/C++</titulo>
  <isbn>9788441526143</isbn>
  <autores><autor>Miguel Angel</autor></autores>
  <paqinas>416</paqinas>
  <editorial>Anaya Multimedia</editorial>
          <año>2016</año>
 </libro>
 libro>
  <titulo>Redes locales</titulo>
  <isbn>9788441519800</isbn>
  <autores><autor>Jim Doherty</autor></autores>
  <paginas>544</paginas>
  <editorial>Rama</editorial>
          <año>2018</año>
 </libro>
 libro>
  <titulo>Lenguajes de Marcas</titulo>
  <isbn>9711441526143</isbn>
  <autores><autor>Miguel Angel</autor></autores>
  <paginas>500</paginas>
  <editorial>Anaya Multimedia</editorial>
          <año>2020</año>
 </libro>
 libro>
  <titulo>Lenguajes WEB</titulo>
  <isbn>9722441526143</isbn>
  <autores><autor>Miguel Angel</autor></autores>
  <paginas>510</paginas>
  <editorial>Paraninfo</editorial>
          <año>2020</año>
 </libro>
 libro>
  <titulo>Gestores de contenido</titulo>
  <isbn>9722441523333</isbn>
  <autores><autor>Miguel Angel</autor></autores>
  <paginas>515</paginas>
  <editorial>Paraninfo</editorial>
          <año>2021</año>
 </libro>
</catalogo>
```

```
<?xml version="1.0" encoding="UTF-8"?>
                                                                                        body{
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                                                                                                 font-family: Verdana;
xmlns="http://www.w3.org/1999/xhtml">
                                                                                                 background: green;
    <xsl:output method="html" encoding="UTF-8" indent="ves"/>
                                                                                                 text-align: center;
  <xsl:template match="/">
    <html>
                                                                                        h1 {
                                                                                                 color: white;
      <head>
         <title>Catalogo</title>
                                                                                                 text-align: center;
        k rel="stylesheet" type="text/css" href="estilo1.css"/>
      </head>
                                                                                        ul{
      <body>
                                                                                                 list-style-position: inside;
         <div>
           <br/><br/>
          <h1>Lista Título y Editorial</h1>
             <xsl:for-each select="/catalogo/libro">
                                                                                        /**Tabla**/
               Título: <xsl:value-of select="titulo"/> - Editorial: <xsl:value-of</li>
select="editorial"/> 
                                                                                        table {
             </xsl:for-each>
                                                                                                 background-color: lightgreen;
           border: 1px solid black;
                                             <br/>
                                                                                                 margin: auto;
           <h1>Tabla con el título, autor y editorial</h1>
           TítuloAutorEditorial
           <xsl:for-each select="/catalogo/libro">
                                                                                        td {
           <xsl:value-of</td><xsl:value-of</td>
                                                                                                 border: 1px solid black;
select="autores/autor"/><xsl:value-of select="editorial"/>
           </xsl:for-each>
                                                                                                 text-align: center;
           <br>></br>
         </div>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```

# Ejemplo 1B

Construye una plantilla XSL y transforma el documento EJ1.xml en un documento HTML que contenga la siguiente información:

- Lista con el título de los libros y su año de publicación, de los libros publicados en el 2018 o años posteriores.
- Tabla con el título del libro, autores y editorial de los libros del autor Miguel Angel.

Construye un CSS y asígnale al documento html para que tenga una presentación armónica.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
     <xsl:output method="html" encoding="UTF-8" indent="yes"/>
  <xsl:template match="/">
    <html>
       <head>
         <title>Catalogo</title>
         k rel="stylesheet" type="text/css" href="estilo1.css"/>
       </head>
       <body>
         <div>
           <br/><br/>
           <h1>Lista Título y Editorial</h1>
           <xsl:for-each select="/catalogo/libro[año>=2018]">
               Título: <xsl:value-of select="titulo"/> - Año: <xsl:value-of select="año"/> 
             </xsl:for-each>
           <br/>br/>
           <h1>Tabla con el título, autor y editorial</h1>
           TítuloAutorEditorial
            <xsl:for-each select="/catalogo/libro[autores/autor='Miguel Angel']">
            <xsl:value-of select="titulo"/><xsl:value-of select="autores/autor"/><xsl:value-of select="editorial"/>
            </xsl:for-each>
           <br></br>
         </div>
       </body>
    </html>
  </xsl:template>
■/xsl:stylesheet>
```

## Ejemplo 2

Construye una plantilla XSL y transforma el documento EJ2.xml en un documento HTML que contenga la siguiente información:

- Lista con el nombre y la raza de los perros cuyo id igual a P02 o igual a P04.
- Tabla con el nombre raza y peso de los perros con un peso superior a 10 kg..

Construye un CSS y asígnale al documento html para que tenga una presentación armónica.

```
<?xml version="1.0" encoding="UTF-8"?>
orectora>
<perros>
 <perro id="P01">
  <nombre>Yogui</nombre>
  <raza>Fox Terrier</raza>
  <peso>8</peso>
 </perro>
 <perro id="P02">
  <nombre>Visi</nombre>
  <raza>Fox Terrier</raza>
  <peso>5</peso>
 </perro>
 <perro id="P03">
  <nombre>Marla</nombre>
  <raza>American Stanford</raza>
  <peso>13</peso>
 </perro>
 <perro id="P04">
  <nombre>Laika</nombre>
  <raza>Fox Terrier</raza>
  <peso>7</peso>
 </perro>
 <perro id="P05">
  <nombre>Bepo</nombre>
  <raza>schnauzer</raza>
  <peso>28</peso>
 </perro>
</perros>
</protectora>
```

```
<?xml version="1.0" encoding="UTF-8"?>
                                                        <?xml version="1.0" encoding="UTF-8"?>
                                                        <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
<?xml-stylesheet type="text/xsl" href="plantilla1.xsl"?>
                                                        xmlns="http://www.w3.org/1999/xhtml">
ctora>
                                                          <xsl:output method="html" encoding="UTF-8" indent="yes"/>
<perros>
  <perro id="P01">
   <nombre>Yogui</nombre>
                                                          <xsl:template match="/">
  <raza>Fox Terrier</raza>
                                                            <html>
   <peso>8</peso>
  </perro>
                                                               <head>
                                                                 <title>Protectora</title>
  <perro id="P02">
                                                                 k rel="stylesheet" type="text/css" href="estilo1.css"/>
  <nombre>Visi</nombre>
                                                               </head>
  <raza>Fox Terrier</raza>
                                                               <body>
  <peso>5</peso>
                                                                 <div>
                                                                   <hr/><hr/>
  </perro>
                                                                   <h1>Lista Nombre y raza</h1>
  <perro id="P03">
  <nombre>Marla</nombre>
                                                                     <xsl:for-each select="/protectora/perros/perro[@id='P02' or @id='P04']">
                                                                       Nombre: <xsl:value-of select="nombre"/> - Raza: <xsl:value-of select="raza"/> 
  <raza>American Stanford</raza>
  <peso>13</peso>
                                                                     </xsl:for-each>
                                                                   </perro>
                                                                                                     <br/>
  <perro id="P04">
                                                                   <h1>Tabla con el nombre, raza y peso</h1>
  <nombre>Laika</nombre>
                                                                   <raza>Fox Terrier</raza>
                                                                    NombreRazaPeso
   <peso>7</peso>
                                                                   <xsl:for-each select="/protectora/perros/perro[peso>10]">
                                                                   <xsl:value-of select="nombre"/><xsl:value-of
  </perro>
                                                        select="raza"/><xsl:value-of select="peso"/>
  <perro id="P05">
                                                                   </xsl:for-each>
  <nombre>Bepo</nombre>
                                                                   <raza>schnauzer</raza>
   <pe><peso>28</peso>
                                                                   <br>></br>
  </perro>
                                                                 </div>
                                                              </body>
                                                            </html>
</protectora>
                                                          </xsl:template>
                                                        </xsl:stvlesheet>
```