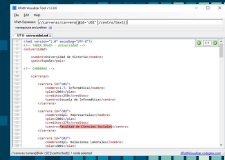


## UT 5 -- Conversión y adaptación de documentos XPath



LMSGI  
DAM1

Julia Triana -- 2020/2021



## INTRODUCCIÓN



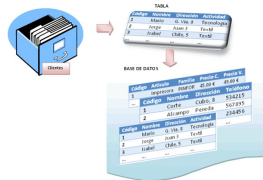
**XML** Se creó XSL (XML Stylesheet Language). Se dividió en:

- XSLT: Transformación de documentos XML
- XPath: Especificación de caminos y expresiones XML
- XSL-FO: Objetos de formateo

**CSS** Estilos y puede usarse con XML



## BASES DE DATOS Y SGBD



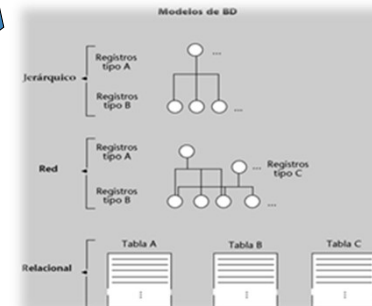
Una base de datos es una aplicación que permite **almacenar, organizar y gestionar información**.

**SGBD**



## Arquitectura de los SGBBDD. DBTG / CODASYL

- Data Base Task Group
- Conference/Committee on Data Systems Languages, on Data System Languages



## \*.XML Y LAS BASES DE DATOS



Trabajando con campos LOB, CLOB ...

Mapeo == BBDD XML enabled

Bases de datos XML nativas



## TRABAJANDO CON CAMPOS LOB, CLOB, NLOB... (I)

Trabajando con LOB, CLOB ...

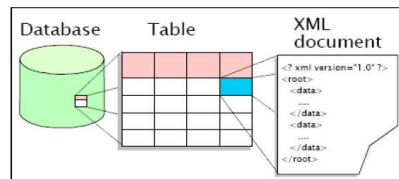
IBM IBM Knowledge Center

Description	Limit
Maximum length of CHAR (in bytes or OCTETS)	254
Maximum length of CHAR (in CODEUNITS32)	63
Maximum length of VARCHAR (in bytes or OCTETS) <sup>2</sup>	32 672
Maximum length of VARCHAR (in CODEUNITS32)	8 168
Maximum length of LONG VARCHAR (in bytes) <sup>1</sup>	32 700
Maximum length of CLOB (in bytes or OCTETS)	2 147 483 647
Maximum length of CLOB (in CODEUNITS32)	536 870 911
Maximum length of serialized XML (in bytes)	2 147 483 647
Maximum length of GRAPHIC (in double-byte characters or CODEUNITS16)	127
Maximum length of GRAPHIC (in CODEUNITS32)	63
Maximum length of VARGRAPHIC (in double-byte characters or CODEUNITS16) <sup>2</sup>	16 336
Maximum length of VARGRAPHIC (in CODEUNITS32)	8 168
Maximum length of LONG VARGRAPHIC (in double-byte characters) <sup>1</sup>	16 350
Maximum length of DBCLOB (in double-byte characters or CODEUNITS16)	1 073 741 823
Maximum length of DBCLOB (in CODEUNITS32)	536 870 911
Maximum length of BLOB (in bytes)	2 147 483 647



## TRABAJANDO CON CAMPOS LOB, CLOB, NLOB... (II)

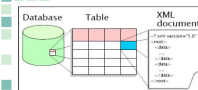
Directamente sobre un campo



La opción de almacenar el documento entero sobre un campo CLOB en una tabla en la base de datos tiene la ventaja que se mantiene el formato del documento, pero el gran inconveniente de no poder realizar en principio ninguna operación de consulta sobre su contenido.



## TRABAJANDO CON CAMPOS LOB, CLOB ... (III)



Creación del campo

```
create table colegio (
  campoclob clob
)
/

insert into colegio values ('
<!--?xml version="1.0" encoding="ISO-8859-1"?-->

<xml>
  <colegio>
    <alumnos>
      <alumno>
        <id_alumno>1</id_alumno>
      </alumno>
    </alumnos>
  </colegio>
</xml>
')
```

Manejo del campo

### 52 DBMS\_LOB

The **DBMS\_LOB** package provides subprograms to operate on **BLOBs**, **CLOBs**, **NCLOBs**, **FILEs**, and temporary **LOBs**. You can use **DBMS\_LOB** to access and manipulation specific parts of a **LOB** or complete **LOBs**.



## MAPEO == BBDD XML ENABLED

```
<?xml version="1.0"?>
<dataset>
  <row>
    <nombre>vibora</nombre>
    <categoria>reptil</categoria>
  </row>
  <row>
    <nombre>rana</nombre>
    <categoria>anfibio</categoria>
  </row>
  <row>
    <nombre>tuna</nombre>
    <categoria>pescado</categoria>
  </row>
  <row>
    <nombre>mapache</nombre>
    <categoria>mamifero</categoria>
  </row>
</dataset>
```

Nombre	Categoria
vibora	reptil
rana	anfibio
tuna	pescado
mapache	mamifero

- OpenInsight
- Sybase
- Hibernate
- Oracle



## BASES DE DATOS XML NATIVAS

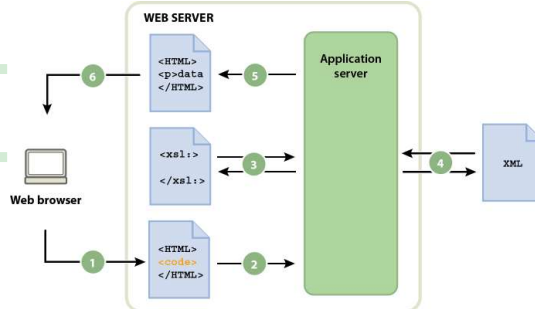
```
<users>
  <user usernombre="a" password="pa"/>
  <user usernombre="b" password="pb"/>
  <user usernombre="c" password="pc"/>
</users>
```

USU. ID	USU. NOMBRE	USU. CONTRA	USU. TIPO
1	VIBORA	1234	REPTIL
2	RANA	5678	ANFIBIO
3	TUNA	9012	PESCAO
4	MAPACHE	3456	MAMIFERO

- XQEngine
- eXist
- Xindice



## XSL -- FAMILIA DE LENGUAJES

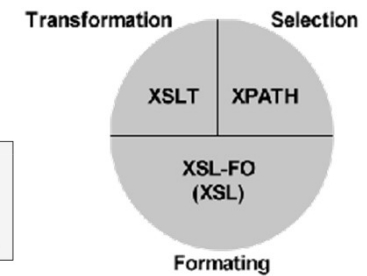
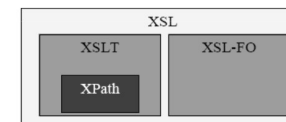


XSL stands for EXtensible Stylesheet Language, and is a style sheet language for XML documents.



## XSL -- FAMILIA DE LENGUAJES

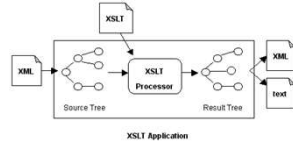
- XPATH - XML Path Language (not XML)
- XSL-FO - XSL Formatting Objects
- XSLT - XSL Transformations



- XPATH -XML Path Language (not XML)
- XSL-FO - XSL Formatting Objects
- XSLT - XSL Transformations

### XSL FAMILIA DE LENGUAJES

- XSLT es la parte más importante de XSL. Usada para transformar un documento XML en otro documento XML, HTML, WML, etc.
- XSLT puede añadir nuevos elementos o incluso eliminarlos. Incluso hacer pruebas o tomar decisiones.
- Comúnmente nos referimos a un XML **source tree** transformado en un XML **result tree**.



- XPATH -XML Path Language (not XML)
- XSL-FO - XSL Formatting Objects
- XSLT - XSL Transformations

### XSL FAMILIA DE LENGUAJES

```

<xsl:stylesheet version = '1.0' xmlns:xsl='http://www.w3.org/1999
<xsl:template match="/">
  <h1>
    <xsl:value-of select="//titulo"/>
  </h1>
  <h2>
    <xsl:value-of select="//autor"/>
  </h2>
</xsl:template>
</xsl:stylesheet>

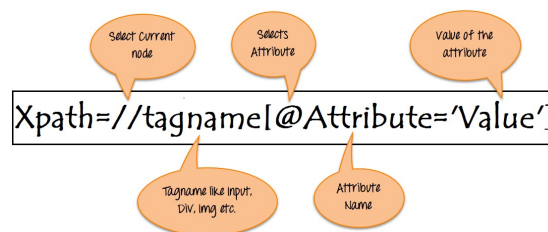
```

- |                              | CSS | XSL |
|------------------------------|-----|-----|
| • ¿Puede usarse con HTML?    | Si  | No  |
| • ¿Puede usarse con XML?     | Si  | Si  |
| • ¿Tiene/Usa transformación? | No  | Si  |
| • Sintaxis utilizada         | CSS | XML |



- XPATH -XML Path Language (not XML)
- XSL-FO - XSL Formatting Objects
- XSLT - XSL Transformations

### XSL -- FAMILIA DE LENGUAJES



### EL LENGUAJE XPATH

```

<?xml version="1.0" encoding="UTF-8"?>
<existencias>
  <producto codigo="234F">
    <material>cobre</material>
    <medida unidades="cm">2</medida>
    <grosor unidades="pulgadas">0.3</grosor>
    <ubicacion>Pale13</ubicacion>
  </producto>
  <producto codigo="124H">
    <material>aluminio</material>
    <medida unidades="cm">3</medida>
    <grosor unidades="pulgadas">0.09</grosor>
    <ubicacion>Pale14</ubicacion>
  </producto>
  <producto codigo="342X">
    <material>hierro</material>
    <medida unidades="cm">4</medida>
    <grosor unidades="pulgadas">0.2</grosor>
    <ubicacion>Pale13</ubicacion>
  </producto>
</existencias>

```

XPath considera un documento XML como un árbol de nodos.



### EL LENGUAJE XPATH: DIRECCIONAMIENTO

Existencias

```
<?xml version="1.0" encoding="UTF-8"?>
<existencias>
  <producto codigo="234F">
    <material>cobre</material>
    <medida unidades="cm">2</medida>
    <grosor unidades="pulgadas">0.3</grosor>
    <ubicacion>Pale13</ubicacion>
  </producto>
  <producto codigo="124H">
    <material>aluminio</material>
    <medida unidades="cm">3</medida>
    <grosor unidades="pulgadas">0.09</grosor>
    <ubicacion>Pale14</ubicacion>
  </producto>
  <producto codigo="342X">
    <material>hierro</material>
    <medida unidades="cm">4</medida>
    <grosor unidades="pulgadas">0.2</grosor>
    <ubicacion>Pale13</ubicacion>
  </producto>
</existencias>
```

Evaluar XPath

XPath: `existencias/producto`

OK Cancel

codigo="234F"  
codigo="124H"  
codigo="342X"  
codigo="234F"  
codigo="789S"

### TIPOS DE NODOS

En XPath no sólo los elementos son nodos.

Los **nodos** del árbol pueden ser:

- ◊ Nodo raíz
- ◊ Nodo elemento
- ◊ Nodo texto
- ◊ Nodo atributo
- ◊ Nodo instrucción de procesamiento
- ◊ Nodo comentario

```
<?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>
```

### TIPOS DE NODOS

```
<?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>
```

nodo raíz  
nodo elemento  
nodo atributo  
nodo de texto

### TIPOS DE RESULTADOS AL EVALUAR EXPRESIONES XPATH

Hay que tener en cuenta que una expresión en XPath no devuelve los elementos que cumplen con el patrón que representa dicha expresión, sino que devuelve una referencia a dichos elementos; es decir, una **expresión XPath** nos devuelve una lista de **apuntadores a los elementos que encajan en el patrón**. Dicha lista puede estar vacía o contener uno o más nodos (**node-set**). En algunos casos, las expresiones XPath devuelven un valor **booleano** o un valor **numérico** o un valor de tipo **cadena**. Para poder evaluar una expresión XPath, el documento XML debe estar bien formado.

Evaluar XPath

XPath: `existencias/producto/@codigo`

OK Cancel

codigo="234F"  
codigo="124H"  
codigo="342X"  
codigo="234F"  
codigo="789S"

Evaluar XPath

XPath: `existencias/producto/@grosor`

OK Cancel

Ningún nodo correspondiente encontrado

## SINTAXIS DE LAS EXPRESIONES XPATH

path en Linux: /usr/home/carrera/asignatura

expresión en XPath: /universidad/carrera/asignatura

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

XPath: child::universidad/child::carreras/\*

XPath: universidad/carreras/\*

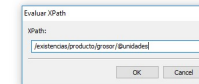
Results from file:///D:/DATOS/IES/Sotro



## LOCALIZACIÓN: RUTA DE LOCALIZACIÓN

el resultado generado siempre es un **conjunto de nodos**.

Se compone de varios pasos separados por /



unidades="pulgadas"  
unidades="pulgadas"  
unidades="pulgadas"  
unidades="pulgadas"  
unidades="cm"

La ruta puede ser:

- Relativa: se evalúa desde el nodo de contexto
- Absoluta: se evalúa desde el nodo raíz (comienza por /)
  - Absolutas: /libro/titulo
  - Relativas: libro/titulo



## LOCALIZACIÓN: PASOS DE LOCALIZACIÓN

Cada paso se compone de:

- Un eje (por defecto es `child`)
- Una prueba de nodo
- Opcionalmente, varios predicados entre []

Sintaxis:

`eje::pruebaNodo[predicado1] [predicado2]...`

XPath Expression: descendant::libro[titulo="La vida está en otra parte"]  
namespaces and prefixes: [g]

ppt\_libreria.xml [x]

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milan Kundera</autor>
6     <fechaPublicacion año="1973"/>
7   </libro>
8   <libro>
9     <titulo>Pantaleón y las visitadoras</titulo>
10    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa

```



## LOCALIZACIÓN: PASOS DE LOCALIZACIÓN

`eje::pruebaNodo[predicado1] [predicado2]...`

Sintaxis **completa**

Sintaxis **abreviada**

XPath Expression: descendant::libro[titulo="La vida está en otra parte"]  
namespaces and prefixes: [g]

ppt\_libreria.xml [x]

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milan Kundera</autor>
6     <fechaPublicacion año="1973"/>
7   </libro>
8   <libro>
9     <titulo>Pantaleón y las visitadoras</titulo>
10    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa

```

/titulo>  
Mario Varg





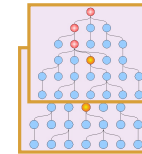
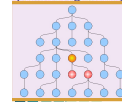
## EXPRESIONES XPATH – SINTAXIS COMPLETA

Nombre del eje	Resultado
ancestor	Selecciona todos los ancestros (padre, abuelo, etc.) del nodo contexto
ancestor-or-self	Selecciona todos los ancestros (padre, abuelo, etc.) del nodo contexto y el propio nodo contexto
attribute	Selecciona todos los atributos del nodo contexto
child	Selecciona todos los hijos del nodo contexto
descendant	Selecciona todos los descendientes (hijos, nietos, etc.) del nodo contexto
descendant-or-self	Selecciona todos los descendientes (hijos, nietos, etc.) del nodo contexto y el propio nodo contexto
following	Selecciona todo lo que haya en el documento después de la etiqueta final del nodo contexto
following-sibling	Selecciona todos los hermanos del nodo contexto que haya tras él
namespace	Selecciona todos los nodos de espacio de nombres del nodo contexto
parent	Selecciona el padre del nodo contexto
preceding	Selecciona todo lo que haya en el documento antes de la etiqueta inicial del nodo contexto
preceding-sibling	Selecciona todos los hermanos del nodo contexto que haya delante de él
self	Selecciona el nodo contexto



## EXPRESIONES XPATH – SINTAXIS COMPLETA

child Hijos directos del nodo actual  
 parent Padre del nodo actual  
 descendant Descendientes  
 ancestor Antecesores  
 descendant-or-self Descendientes incluido el nodo actual  
 ancestor-or-self Antecesores incluido el nodo actual  
 following Los nodos siguientes (incluidos los descendientes)  
 preceding Los precedentes (excluyendo los antecesores)  
 following-sibling Hermanos siguientes  
 preceding-sibling Hermanos precedentes



XPath Expression: descendant::libro[título="La vida está en otra parte"]



## EXPRESIONES XPATH – SINTAXIS ABREVIADA

```

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

```

/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/a/autor	No devuelve nada porque "autor" no es hijo de "biblioteca".



## EXPRESIONES XPATH

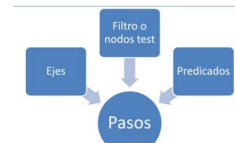
Las expresiones son evaluadas, y pueden devolver 4 tipos distintos de resultados:

- Boolean
- Node-Set
- Number
- String

Cada expresión se compone de los pasos:

- Un eje
- Filtro (nodo set)
- Opcionalmente, varios predicados

XPath Expression: //libro[título="La vida está en otra parte"]

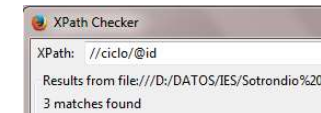
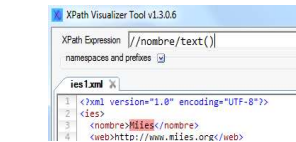
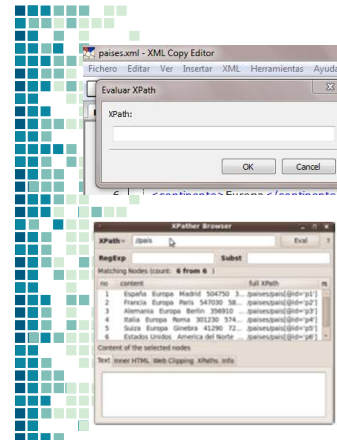


## EXPRESIONES XPATH – SINTAXIS ABREVIADA

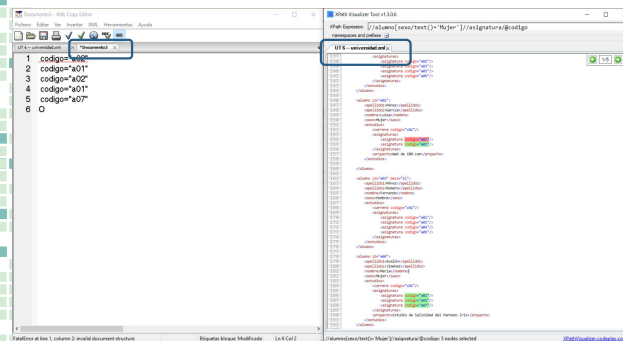
Expression	Description
<i>Nodename</i>	Selects named node
/	Selects from the root node (start of expression) or the current node
//	Selects any descendant of the root node (start of expression) or the current node
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes
*	Wildcard – matches any element
node()	Matches any node of any kind
[ ]	Brackets that contain a predicate, which is used to find a specific node



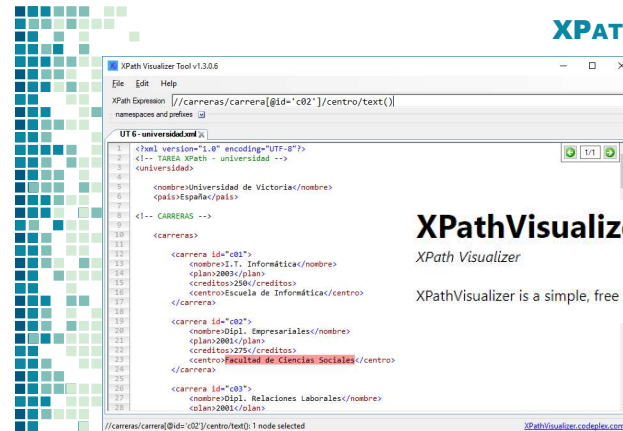
## HERRAMIENTAS PARA EL MANEJO DE XPATH



## HERRAMIENTAS PARA EL MANEJO DE XPATH



## XPATHVISUALIZER



## XPathVisualizer

*XPath Visualizer*

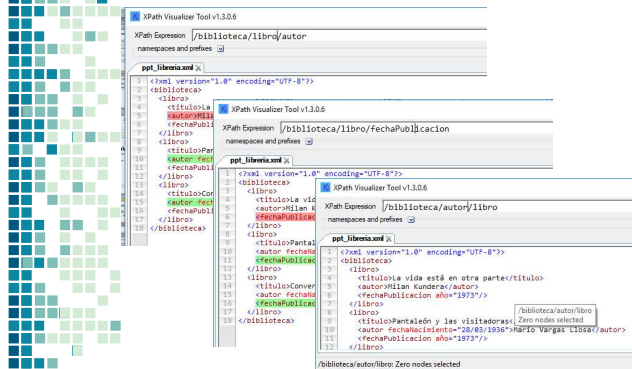
XPathVisualizer is a simple, free Winforms tool





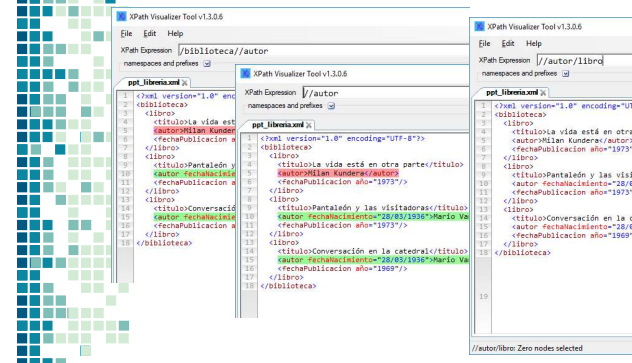
**/:** si está al principio de la expresión, indica el nodo raíz, sino, indica "hijo".

### TRABAJANDO CON XPATH - EJES



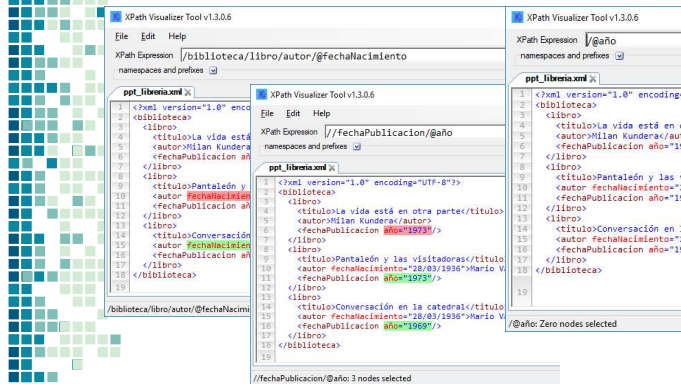
**//:** indica "descendiente" (hijos, hijos de hijos, etc.).

### TRABAJANDO CON XPATH - EJES



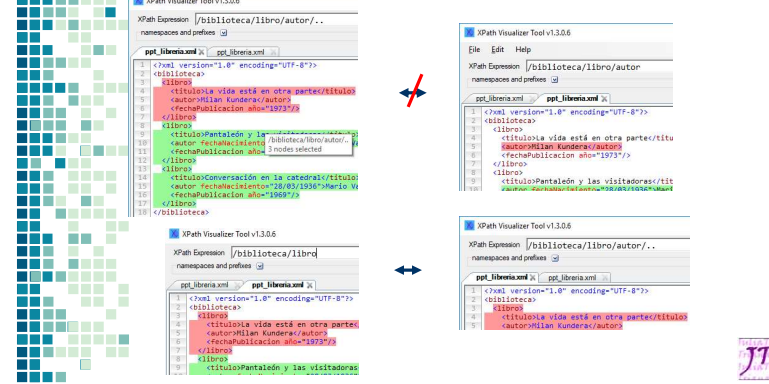
**@atributo:** selecciona el atributo.

### TRABAJANDO CON XPATH - EJES

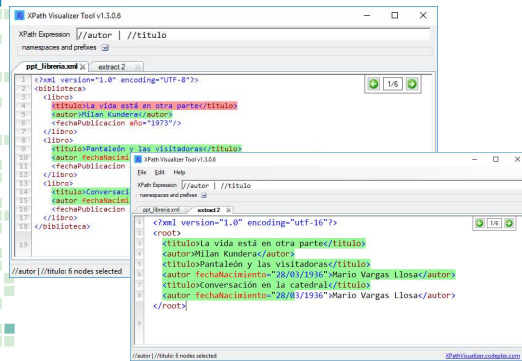


**..:** selecciona el elemento padre.

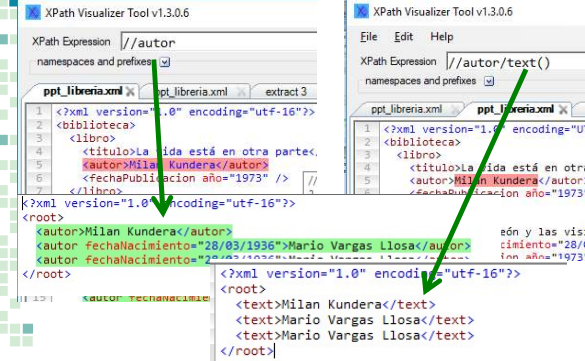
### TRABAJANDO CON XPATH - EJES



l: permite elegir varios recorridos. **TRABAJANDO CON XPATH - EJES**

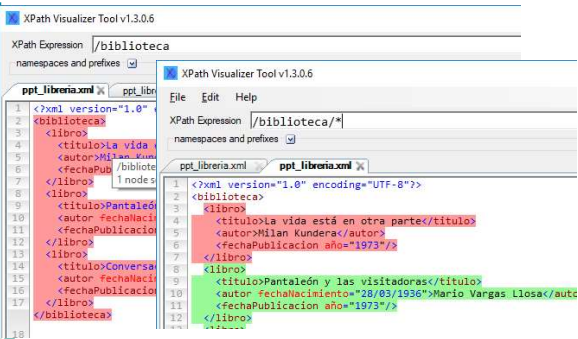


**text() – Contenido del elemento** **XPATH – ACCESO AL CONTENIDO TEXTUAL**



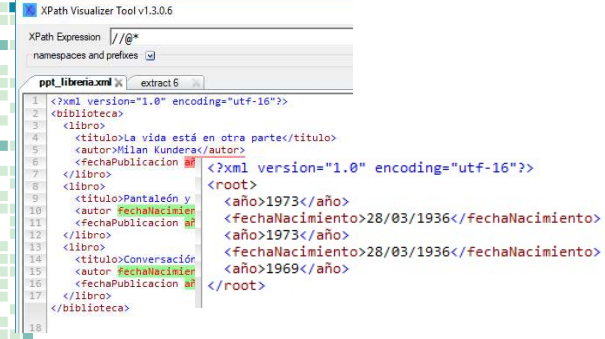
**\*: selecciona todos los elementos**

**XPATH – \***



**@\*: selecciona todos los atributos**

**XPATH – \***



## XPATH – PREDICADOS

Los predicados se escriben entre corchetes

- ✚ **[@atributo]**: selecciona los **elementos que tienen el atributo**.
- ✚ **[número]**: si hay varios resultados selecciona uno de ellos por número de orden; **last()** selecciona el último de ellos
- ✚ **[condición]**: selecciona los nodos que cumplen la condición. La condición puede utilizar el valor de un atributo (utilizando "@" o el texto que contiene el elemento (utilizando el "."))

```
//libro[precio>=24]
```

```
//libro[2]
```



## XPATH - PREDICADOS – ELEMENTOS DEL LENGUAJE

### Operadores

Booleanos	Aritméticos
= Igual	+ Suma
!= Diferente	- Resta
< Menor	* Multiplicación
<= Menor o igual	div División
> Mayor	mod Módulo,
>= Mayor o igual	resto de la
or O	división
and Y	

### Funciones

- Boolean
- Node-Set
- Numeric
- String

Funciones en XPath



## biblioteca.xml

## XPATH – PREDICADOS

```

1 <?xml version="1.0" encoding="ISO-8859-1"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milan Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



## XPATH – PREDICADOS

```

File Edit Help
XPath Expression: //libro[autor!=Mario Vargas Llosa]
namespaces and prefixes
UT 6 - biblioteca_precios.x...
1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milan Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



## TAREA EJEMPLO – LA DE LA BIBLIOTECA CON PRECIOS

- Seleccionar todos los **títulos** (no los elementos).

XPath Expression `//titulo/text()`

namespaces and prefixes

UT 6 – biblioteca\_precios.x..

```

1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milán Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



- Seleccionar el nodo título del **antepenúltimo libro** (el elemento).

XPath Expression `/biblioteca/libro[last()-2]/titulo`

namespaces and prefixes

UT 6 – biblioteca\_precios.x..

```

1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milán Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



- Seleccionar el **precio** de todos los nodos (el texto).

XPath Expression `//libro/precio/text()`

namespaces and prefixes

UT 6 – biblioteca\_precios.x..

```

1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milán Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



- Seleccionar el **precio de los libros con precio > 25** (el elemento)

XPath Expression `//libro[precio>25]/precio`

namespaces and prefixes

UT 6 – biblioteca\_precios.x..

```

1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milán Kundera</autor>
6     <fechaPublicacion anio="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion anio="1973"/>
13    <precio>30</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion anio="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



- Seleccionar el título y el precio de los libros con precio >25 (el texto).

XPath Expression `//libro[precio>25]/titulo/text() | //libro[precio>25]/precio/text()`

namespaces and prefixes ☐

UT 6 - biblioteca\_precios.x - X

```

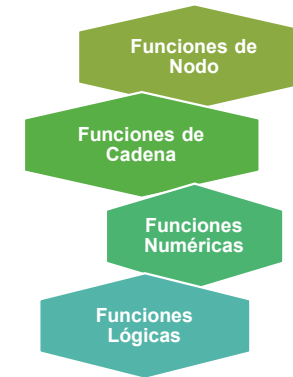
1 <?xml version="1.0" encoding="UT-8"?>
2 <biblioteca>
3   <libro>
4     <titulo>La vida está en otra parte</titulo>
5     <autor>Milán Kundera</autor>
6     <fechaPublicacion año="1973"/>
7     <precio>23.5</precio>
8   </libro>
9   <libro>
10    <titulo>Pantaleón y las visitadoras</titulo>
11    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
12    <fechaPublicacion año="1973"/>
13    <precio>38</precio>
14  </libro>
15  <libro>
16    <titulo>Conversación en la catedral</titulo>
17    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
18    <fechaPublicacion año="1969"/>
19    <precio>28.25</precio>
20  </libro>
21 </biblioteca>

```



www.w3.org/TR/xpath-functions/

## FUNCIONES XPATH



## FUNCIONES DE NODO

- Cuantos libros tenemos de **Milán Kundera**?

XPath: `count(//libro[autor="Milán Kundera"])`

- Nombre del elemento padre de **precio**?

XPath: `name(//precio/..)`

- Contiene el título del último libro la palabra **catedral**?

XPath: `contains(//libro[last()]/titulo/text(),"catedral")`



## FUNCIONES DE CADENA

- Cambia el carácter á por A en el nombre del primer libro.

XPath: `translate(//libro/titulo[1]/text(),"á","A")`

XPath: `concat(//titulo/text()," - ",//autor/text())`

Results from file:///D:/DATOS/IES/Sotondio%2020

Returned a value

1: La vida está en otra parte - Milán Kundera

- Título y autor del libro.

XPath: `substring(//autor/@fechaNacimiento,7,4)`

Results from file:///D:/DATOS/IES/Sotondio%2020

Returned a value

1: 1936

- Año de la fecha de nacimiento.





## FUNCIONES DE NUMÉRICAS

- Desglose del pago.

XPath: `concat(sum(//precio), " + ", sum(//precio)*0.21, " (IVA) ", " = ", sum(//precio)*1.21)`

Returned a value

1: 81.75 + 17.1675 (IVA) = 98.9175

- Cuantos libros se publicaron en 1973?

XPath: `count(//fechaPublicacion[@anio="1973"])`

Results from file:///D:/DATOS/IES/Sotrondio%2020

Returned a value

1: 2



## COMUNIDAD DE VECINOS

- Ruta de acceso al elemento nombre.
- Valor del elemento nombre.
- Ruta de acceso que seleccione las viviendas que hay en el piso 1.

```
<!-- Tarea de aula vecinos -->
<edificio>
  <vivienda>
    <piso>1</piso>
    <puerta>A</puerta>
  </vivienda>
  <vecino>
    <nombre>David</nombre>
    <nombre>Sara</nombre>
  </vecino>
</edificio>
```

XPath: `/edificio/vivienda/vecinos/nombre`

Results from file:///C:/DATOS/Sotrondio

4 matches found

1: <nombre>David</nombre>  
2: <nombre>Sara</nombre>  
3: <nombre>Irene</nombre>  
4: <nombre>Jaime</nombre>

XPath: `/edificio/vivienda/vecinos/nombre/text()`

Results from file:///C:/DATOS/S

4 matches found

1: David  
2: Sara  
3: Irene  
4: Jaime

XPath: `/edificio/vivienda[piso=1]`

Results from file:///C:/DATOS/Sotrondio%20

2 matches found

1: <vivienda>  
 <piso>1</piso>  
 <puerta>A</puerta>  
 <vecinos>  
2: <vivienda>  
 <piso>1</piso>  
 <puerta>B</puerta>  
 <vecinos>



## TICKETS

- Suma los precios.
- Suma el total.

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Documento XML para los ejercicios de la UT 6 -->
<listatickets>
  <ticket>
    <numero>7483</numero>
    <fecha>2018-08-20</fecha>
    <hora>22:00:00</hora>
    <producto>
      <nombre>Agua</nombre>
      <precio>1</precio>
    </producto>
    <producto>
      <nombre>Refresco</nombre>
      <precio>1.5</precio>
    </producto>
    <total>2.5</total>
  </ticket>
  <ticket>
    <numero>7484</numero>
    <fecha>2018-08-25</fecha>
    <hora>22:05:00</hora>
    <producto>
      <nombre>Refresco</nombre>
      <precio>1.5</precio>
    </producto>
    <producto>
      <nombre>Refresco</nombre>
      <precio>1.5</precio>
    </producto>
    <total>3</total>
  </ticket>
</listatickets>
```

XPath: `sum(//listatickets/ticket/producto/precio)`

Results from file:///C:/DATOS/Sotrondio

Returned a value

1: 5.5

XPath: `sum(//listatickets/ticket/total)`

Results from file:///C:/DATOS/Sotrondio

Returned a value

1: 5.5



## PELÍCULAS

- Cuantas son las películas con valoración de 4 o mas puntos.

```
<!-- Tarea aula películas -->
<peliculas>
  <pelicula>
    <nombre>Super 8</nombre>
    <valoracion puntos>"4"/>
    <publico>Adultos</publico>
  </pelicula>
  <pelicula>
    <nombre>Los pitufos</nombre>
    <valoracion puntos>"3"/>
    <publico>Todos los públicos</publico>
  </pelicula>
</peliculas>
```

XPath: `count(/peliculas//valoracion[@puntos=4])`





