
UNIDAD 8

CONVERSIÓN Y ADAPTACIÓN DE DOCUMENTOS XML

LENGUAJES DE MARCA Y SISTEMAS DE GESTIÓN DE INFORMACIÓN

ÍNDICE DE CONTENIDOS

1. Introducción
2. XPath: Conceptos esenciales
 1. Antes de nada
 2. Sintaxis XPath
 3. Operadores XPath
 4. Funciones XPath
 5. Más ejemplos

ÍNDICE DE CONTENIDOS

1. Introducción
2. XPath: Conceptos esenciales
 1. Antes de nada
 2. Sintaxis XPath
 3. Operadores XPath
 4. Funciones XPath
 5. Más ejemplos

I. INTRODUCCIÓN

- XSL (Extensible Stylesheet Language) es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera, debe ser transformada o formateada para su presentación en un medio (pantalla, impresora, etc.). Puede considerarse que XSL es el CSS de XML.
- XSL permite tomar pleno control sobre los datos, pudiendo establecer criterios sobre qué datos ver y en qué orden visualizarlos, estableciendo filtros y definiendo formatos de salida para su presentación.
- Extensible Stylesheet Language (XSL) es una recomendación del W3C:

<http://www.w3.org/TR/xsl/>

I. INTRODUCCIÓN

CSS = Style Sheets for HTML

HTML uses predefined tags, and the meaning of each tag is **well understood**.

The <table> tag in HTML defines a table - and a browser knows **how to display it**.

Adding styles to HTML elements are simple. Telling a browser to display an element in a special font or color, is easy with CSS.

XSL = Style Sheets for XML

XML does not use predefined tags (we can use any tag-names we like), and therefore the meaning of each tag is **not well understood**.

A <table> tag could mean an HTML table, a piece of furniture, or something else - and a browser **does not know how to display it**.

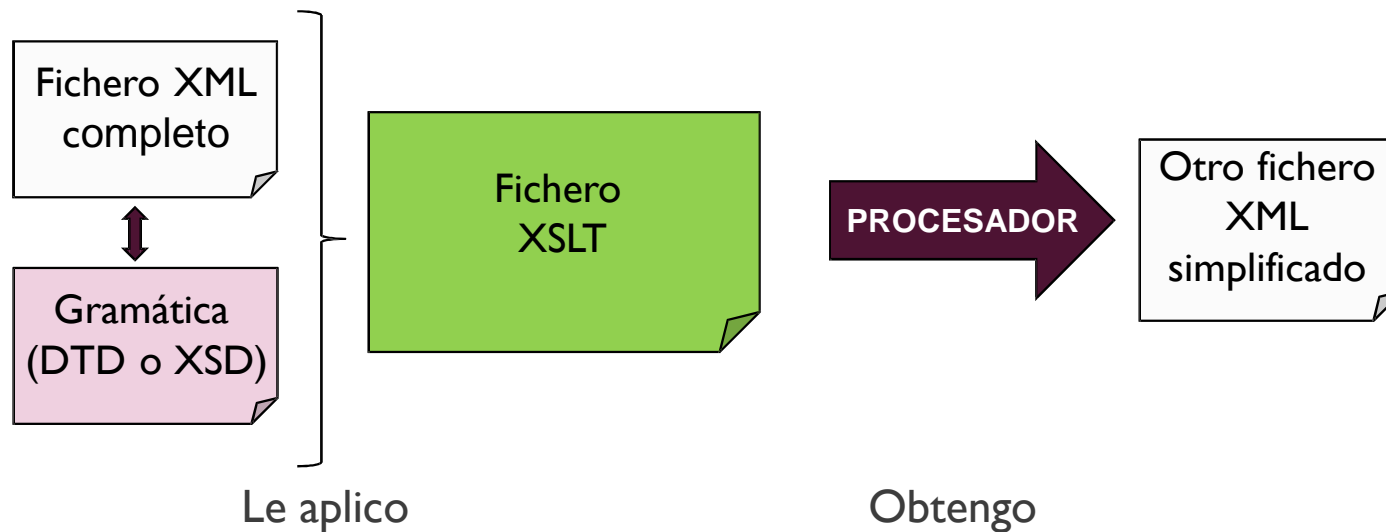
XSL describes how the XML document should be displayed!

I. INTRODUCCIÓN

- XSL es una familia de lenguajes formada por:
 - **XSLT**: permite convertir documentos XML de una sintaxis a otra (por ejemplo, de un XML a otro XML simplificado o de un XML a un documento HTML).
 - **XSL-FO**: permite especificar el formato visual con el cual se quiere presentar un documento XML, es usado principalmente para generar documentos PDF o documento que serán enviado a la impresora directamente.
- XSLT y XSL-FO usan internamente a su vez:
 - **XPath**: permite acceder o referirse a porciones de un documento XML.

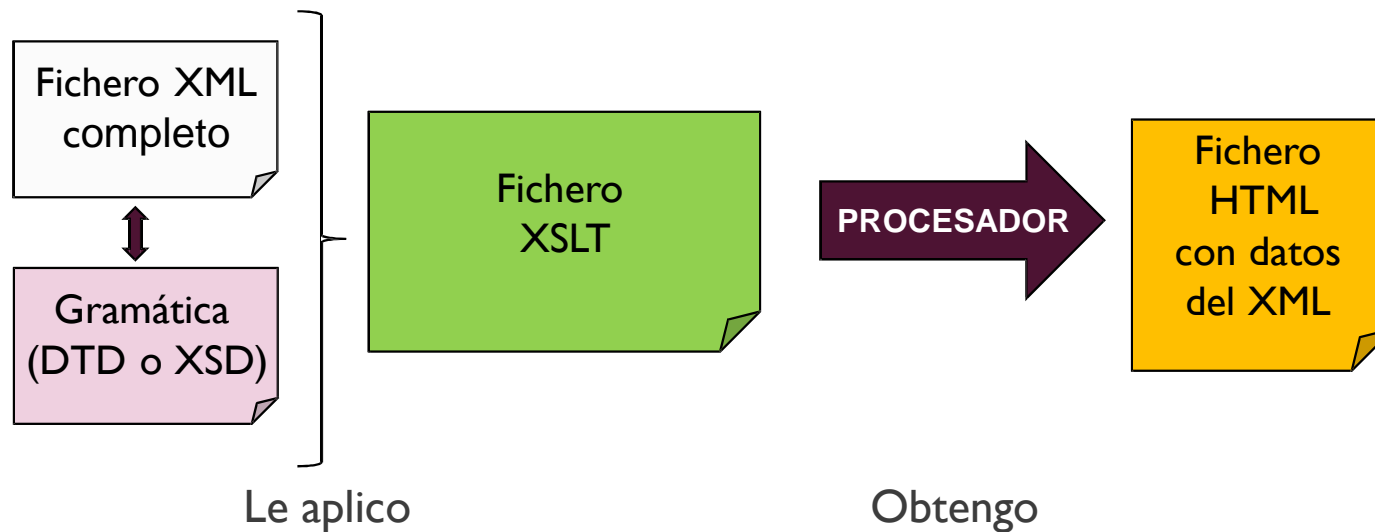
I. INTRODUCCIÓN

- XSLT y XPath: XML Simplificado



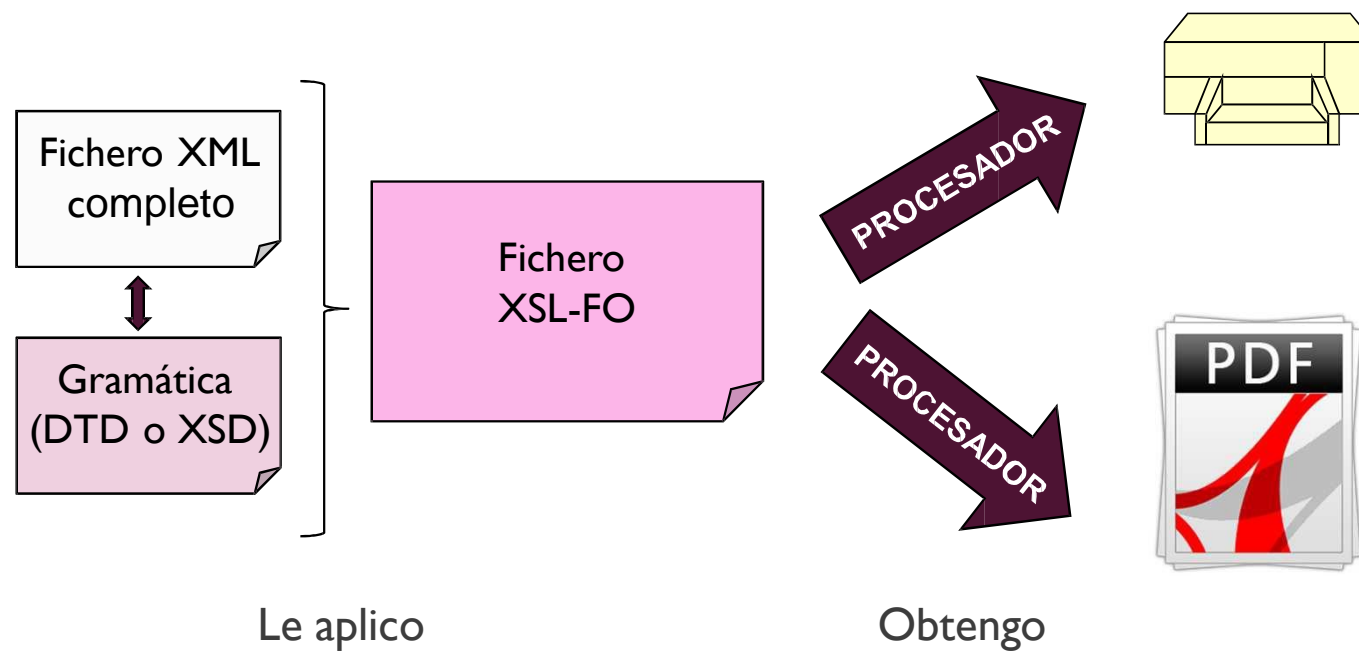
I. INTRODUCCIÓN

- XSLT y XPath: HTML



I. INTRODUCCIÓN

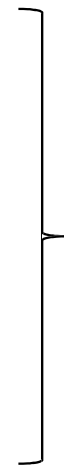
- XSL-FO y XPath: Salida a impresora



I. INTRODUCCIÓN

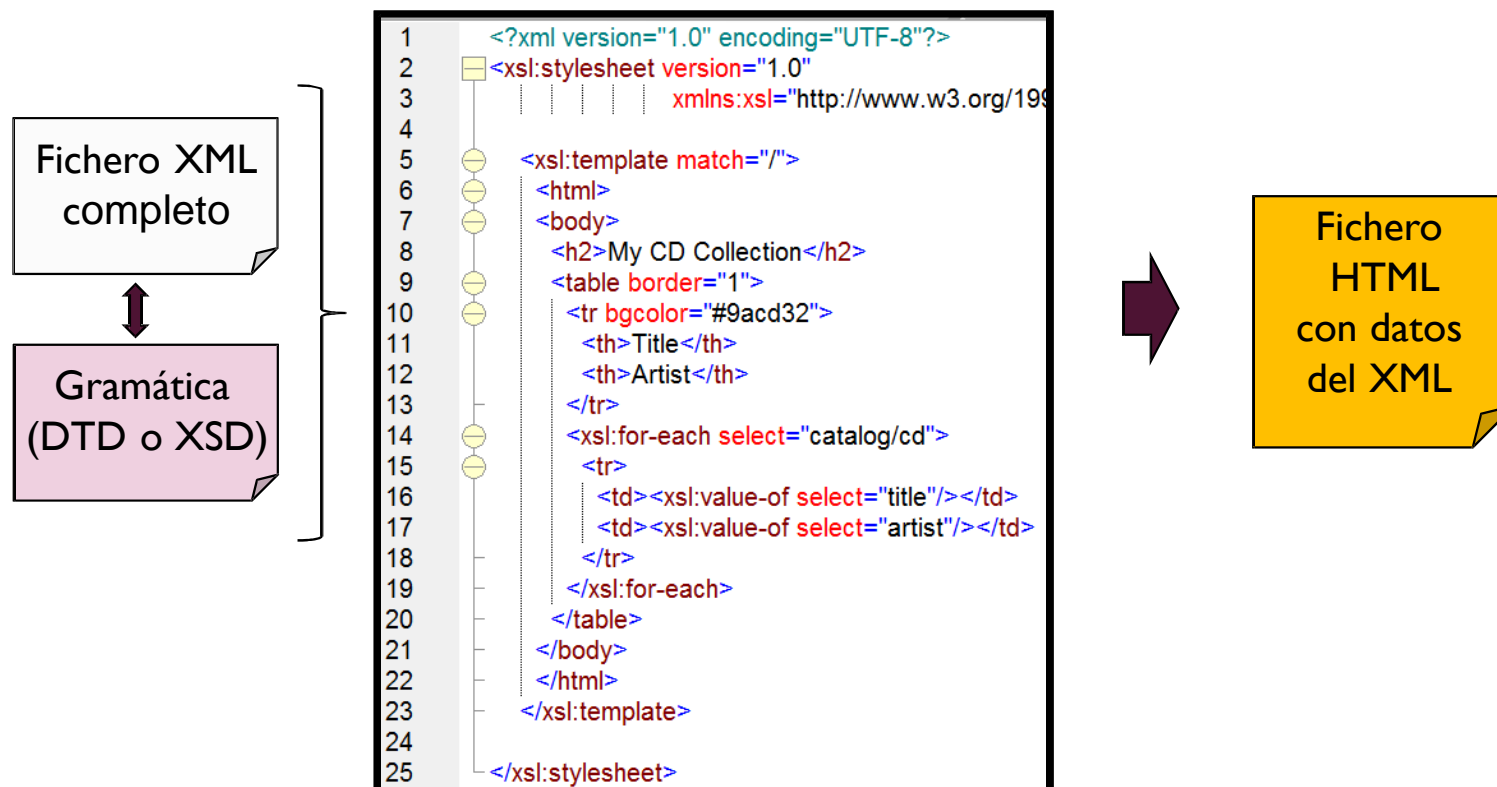
- Ejemplo 1: XSLT y XPath

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE catalog SYSTEM "catalog.dtd">
3  <catalog>
4    <cd>
5      <title>Empire Burlesque</title>
6      <artist>Bob Dylan</artist>
7      <country>USA</country>
8      <company>Columbia</company>
9      <price>10.90</price>
10     <year>1985</year>
11   </cd>
12   <cd>
13     <title>Hide your heart</title>
14     <artist>Bonnie Tyler</artist>
15     <country>UK</country>
16     <company>CBS Records</company>
17     <price>9.90</price>
18     <year>1988</year>
19   </cd>
20   <cd>
21     <title>Greatest Hits</title>
22     <artist>Dolly Parton</artist>
23     <country>USA</country>
24     <company>RCA</company>
25     <price>9.90</price>
26     <year>1982</year>
27   </cd>
```



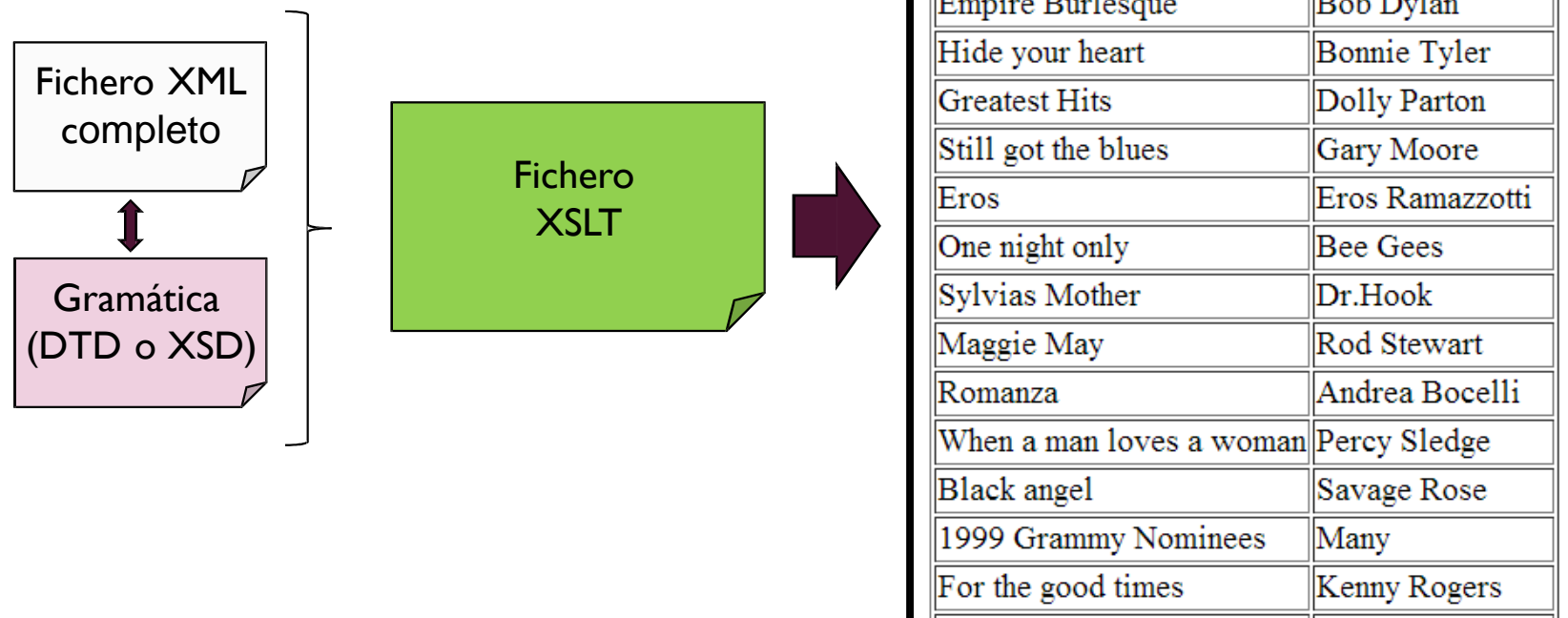
I. INTRODUCCIÓN

- Ejemplo I: XSLT y XPath



I. INTRODUCCIÓN

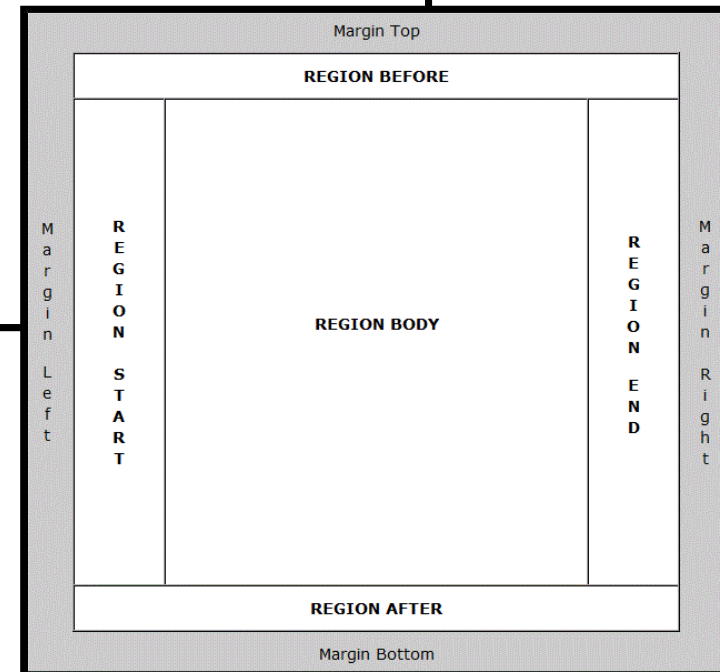
- Ejemplo 1: XSLT y XPath



I. INTRODUCCIÓN

- Ejemplo 2: XSL-FO y XPath

```
1 <fo:simple-page-master master-name="A4" page-width="297mm"
2   page-height="210mm" margin-top="1cm" margin-bottom="1cm"
3   margin-left="1cm" margin-right="1cm">
4     <fo:region-body margin="3cm"/>
5     <fo:region-before extent="2cm"/>
6     <fo:region-after extent="2cm"/>
7     <fo:region-start extent="2cm"/>
8     <fo:region-end extent="2cm"/>
9 </fo:simple-page-master>
```



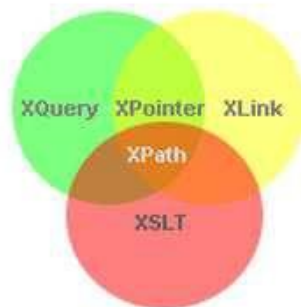
ÍNDICE DE CONTENIDOS

1. Introducción
2. XPath: Conceptos esenciales
 1. Antes de nada
 2. Sintaxis XPath
 3. Operadores XPath
 4. Funciones XPath
 5. Más ejemplos

2. XPATH: CONCEPTOS ESENCIALES

- XPath es un lenguaje de expresión utilizado para referenciar nodos de información en un conjunto de datos XML.
- Con XPath podremos seleccionar y hacer referencia a texto, elementos, atributos y cualquier otra información contenida dentro de un fichero XML.
- XPath por sí solo no tiene "sentido". Siempre se usa en combinación con otra tecnología. Nosotros lo usaremos con XSLT y XQuery.

What is XPath?



- XPath is a syntax for defining parts of an XML document
- XPath uses path expressions to navigate in XML documents
- XPath contains a library of standard functions
- XPath is a major element in XSLT
- XPath is a W3C recommendation

2. XPATH: CONCEPTOS ESENCIALES

- XPath es una recomendación del W3C:

<http://www.w3.org/TR/xpath/>

- Existen varias versiones de XPath:
 - noviembre de 1999: XML Path Language (XPath) 1.0
 - enero de 2007: XML Path Language (XPath) 2.0
 - diciembre de 2010: XML Path Language (XPath) 2.0 (2º edición)
 - abril de 2014: XML Path Language (XPath) 3.0
 - marzo de 2017: XML Path Language (XPath) 3.1

2. XPATH: CONCEPTOS ESENCIALES

2.1. Antes de nada

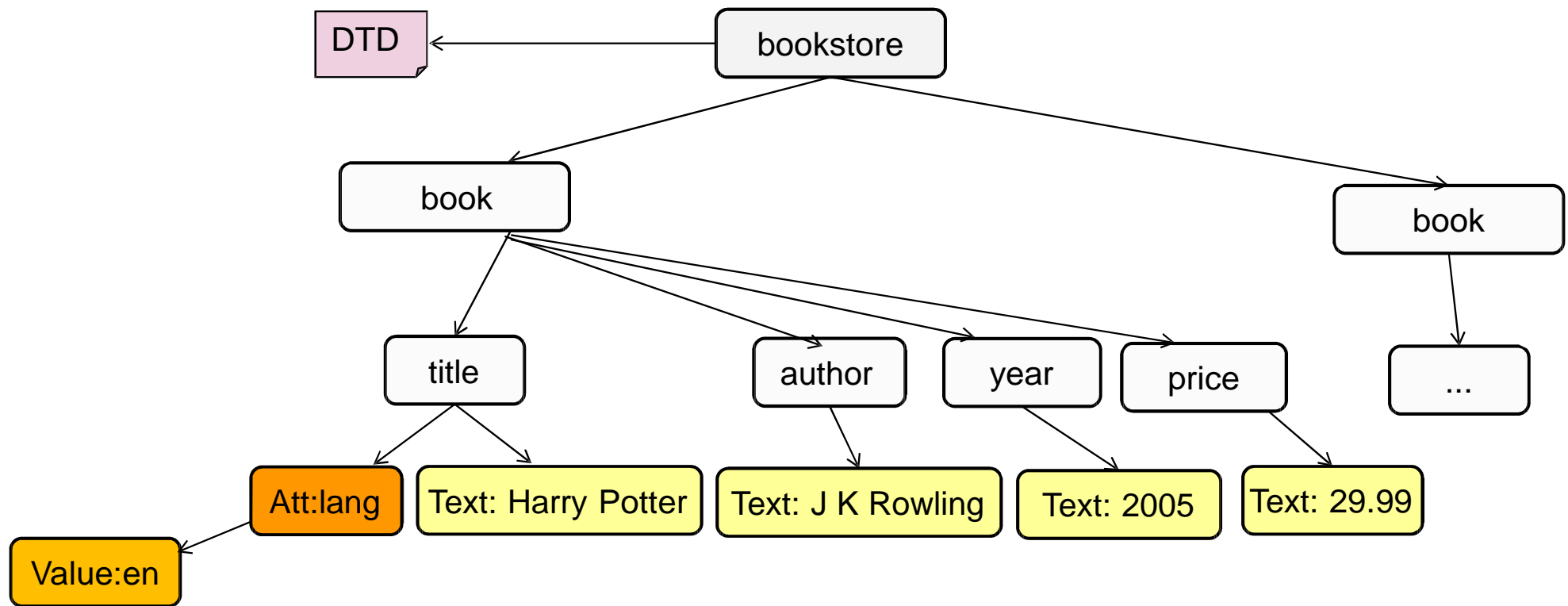
- Dado el siguiente fichero XML:

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd">
3  <bookstore>
4    <book>
5      <title lang="en">Harry Potter</title>
6      <author>J K Rowling</author>
7      <year>2005</year>
8      <price>29.99</price>
9    </book>
10   <book>
11     <title lang="en">LOR</title>
12     <author>J R R Tolkien</author>
13     <year>2004</year>
14     <price>50.55</price>
15   </book>
16 </bookstore>
```

2. XPATH: CONCEPTOS ESENCIALES

2.1. Antes de nada

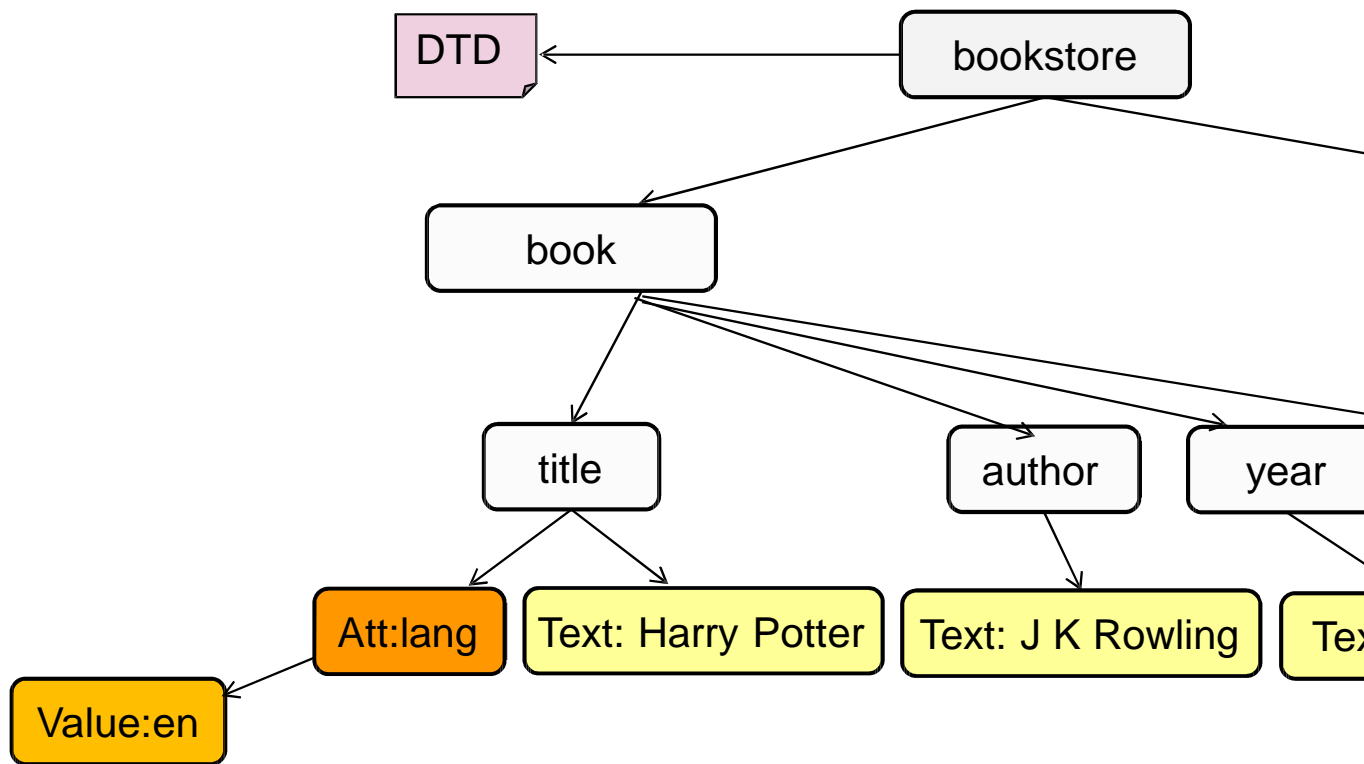
- Dicho fichero XML genera el siguiente árbol:



2. XPATH: CONCEPTOS ESENCIALES

2.1. Antes de nada

- Dicho fichero XML genera el siguiente árbol:



CONCEPTOS:

- ✓ Nodo raíz
- ✓ Nodos elemento
- ✓ Nodos atributos
- ✓ Valores atómicos (text y value)
- ✓ Padre
- ✓ Hijo
- ✓ Hermanos
- ✓ ...

¡¡XPath trabaja con nodos!!

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis XPath

- Selección de nodos

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

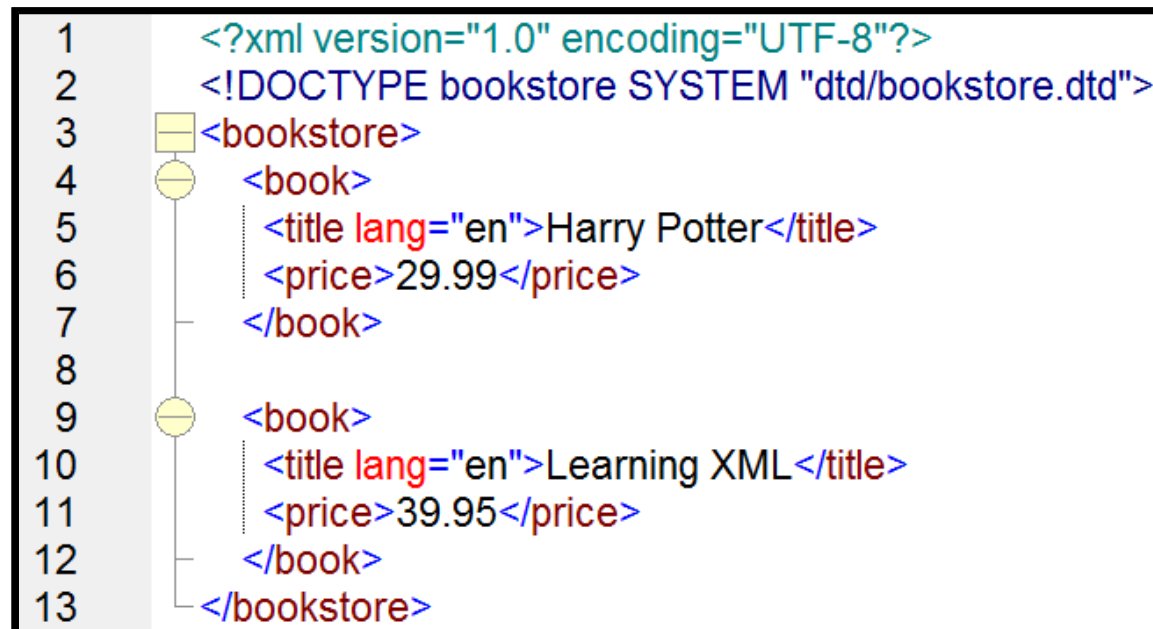
Nota: No confundir la raíz / con el nodo raíz.

El nodo raíz cuelga de / y por tanto tiene un padre, la raíz /.

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis XPath

- Dado el siguiente fichero XML:



The image shows an XML document with 13 lines of code. To the left of the code, a tree diagram illustrates the document structure. The root node is a yellow square labeled 'bookstore' on line 3. It has two child nodes, both represented by yellow circles labeled 'book'. The first 'book' node is on line 4 and contains two child elements: 'title' (line 5) with the value 'Harry Potter' and 'price' (line 6) with the value '29.99'. The second 'book' node is on line 9 and contains two child elements: 'title' (line 10) with the value 'Learning XML' and 'price' (line 11) with the value '39.95'. The tree diagram uses vertical lines to connect the nodes, with horizontal lines indicating the level of each node in the hierarchy.

```
1      <?xml version="1.0" encoding="UTF-8"?>
2      <!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd">
3      <bookstore>
4      <book>
5          <title lang="en">Harry Potter</title>
6          <price>29.99</price>
7      </book>
8
9      <book>
10         <title lang="en">Learning XML</title>
11         <price>39.95</price>
12     </book>
13 </bookstore>
```

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis XPath

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd">
3  <bookstore>
4    <book>
5      <title lang="en">Harry Potter</title>
6      <price>29.99</price>
7    </book>
8
9    <book>
10     <title lang="en">Learning XML</title>
11     <price>39.95</price>
12   </book>
13 </bookstore>
```

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis XPath

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd">
3  <bookstore>
4  <book>
5  <title lang="en">Harry Potter</title>
6  <price>29.99</price>
7  </book>
8
9  <book>
10 <title lang="en">Learning XML</title>
11 <price>39.95</price>
12 </book>
13 </bookstore>
```

Las rutas pueden ser:

- Absolutas:
- Relativas:

book	Selects all nodes with the name "book"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- **Predicados:** Se utilizan para encontrar un nodo específico o un nodo que contiene un valor específico. Siempre están incrustados entre corchetes.

Path Expression	Result
/bookstore/book[1]	Selects the first book element that is the child of the bookstore element. Note: In IE 5,6,7,8,9 first node is [0], but according to W3C, it is [1]. To solve this problem in IE, set the SelectionLanguage to XPath: <i>In JavaScript:</i> <code>xml.setProperty("SelectionLanguage","XPath");</code>
/bookstore/book[last()]	Selects the last book element that is the child of the bookstore element
/bookstore/book[last()-1]	Selects the last but one book element that is the child of the bookstore element
/bookstore/book[position()<3]	Selects the first two book elements that are children of the bookstore element

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Predicados

1	<code><?xml version="1.0" encoding="UTF-8"?></code>
2	<code><!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd"></code>
3	<code><bookstore></code>
4	<code><book></code>
5	<code><title lang="en">Harry Potter</title></code>
6	<code><price>29.99</price></code>
7	<code></book></code>
8	
9	<code><book></code>
10	<code><title lang="en">Learning XML</title></code>
11	<code><price>39.95</price></code>
12	<code></book></code>
13	<code></bookstore></code>

Path Expression
<code>/bookstore/book[1]</code>
<code>/bookstore/book[last()]</code>
<code>/bookstore/book[last()-1]</code>
<code>/bookstore/book[position()<3]</code>

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Predicados

<code>//title[@lang]</code>	Selects all the title elements that have an attribute named lang
<code>//title[@lang='en']</code>	Selects all the title elements that have a "lang" attribute with a value of "en"
<code>/bookstore/book[price>35.00]</code>	Selects all the book elements of the bookstore element that have a price element with a value greater than 35.00
<code>/bookstore/book[price>35.00]/title</code>	Selects all the title elements of the book elements of the bookstore element that have a price element with a value greater than 35.00

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Predicados

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE bookstore SYSTEM "dtd/bookstore.dtd">
3  <bookstore>
4  <book>
5      <title lang="en">Harry Potter</title>
6      <price>29.99</price>
7  </book>
8
9  <book>
10     <title lang="en">Learning XML</title>
11     <price>39.95</price>
12 </book>
13 </bookstore>
```

//title[@lang]

//title[@lang='en']

/bookstore/book[price>35.00]

/bookstore/book[price>35.00]/title

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

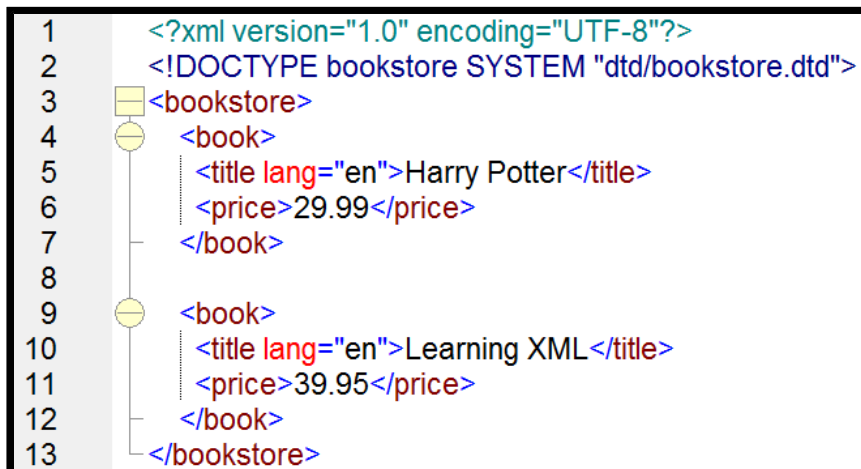
- Selección de nodos desconocidos

Wildcard	Description
*	Matches any element node
@*	Matches any attribute node
node()	Matches any node of any kind

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Selección de nodos desconocidos



Path Expression	Result
/bookstore/*	Selects all the child element nodes of the bookstore element
//*	Selects all elements in the document
//title[@*]	Selects all title elements which have at least one attribute of any kind

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

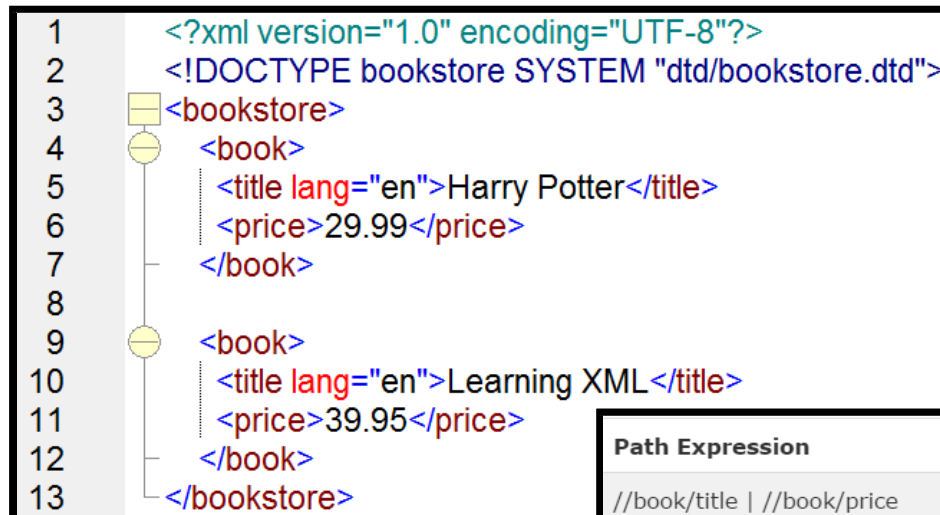
- Selección de varios elementos

Path Expression	Result
<code>//book/title //book/price</code>	Selects all the title AND price elements of all book elements
<code>//title //price</code>	Selects all the title AND price elements in the document
<code>/bookstore/book/title //price</code>	Selects all the title elements of the book element of the bookstore element AND all the price elements in the document

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Selección de varios elementos



Path Expression	Result
<code>//book/title //book/price</code>	Selects all the title AND price elements of all book elements
<code>//title //price</code>	Selects all the title AND price elements in the document
<code>/bookstore/book/title //price</code>	Selects all the title elements of the book element of the bookstore element AND all the price elements in the document

2. XPATH: CONCEPTOS ESENCIALES

2.2. Sintaxis Xpath

- Otros

Existen otros selectores de nodos más avanzados (XPath Axes) que permiten navegar por el árbol y realizar selecciones más avanzadas. Los puedes encontrar en:

http://www.w3schools.com/xpath/xpath_axes.asp

2. XPATH: CONCEPTOS ESENCIALES

2.3. Operadores Xpath

- Algunos de los operadores que usaremos en clase:

=	Equal	price=9.80
!=	Not equal	price!=9.80
<	Less than	price<9.80
<=	Less than or equal to	price<=9.80
>	Greater than	price>9.80
>=	Greater than or equal to	price>=9.80

Más operadores en: http://www.w3schools.com/xpath/xpath_operators.asp

2. XPATH: CONCEPTOS ESENCIALES

2.4. Funciones XPath

- XPath también define una serie de funciones muy útiles, aunque estas funciones las usaremos principalmente cuando estudiemos XQuery

Functions Reference

- [Accessor](#)
- [Error and Trace](#)
- [Numeric](#)
- [String](#)
- [AnyURI](#)
- [Boolean](#)
- [Duration/Date/Time](#)
- [QName](#)
- [Node](#)
- [Sequence](#)
- [Context](#)



The default prefix for the function namespace is fn:

The URI of the function namespace is: <http://www.w3.org/2005/xpath-functions>

- http://www.w3schools.com/xpath/xpath_functions.asp

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos seleccionar todos los nodos `<h1>` que son hijos de un nodo `<body>` que, a su vez, es hijo de un nodo `<html>` que es el nodo raíz del árbol XML. ¿Cómo lo hacemos?
- Queremos seleccionar todos los nodos `<h1>` que aparezcan en cualquier posición del árbol XML. ¿Cómo lo hacemos?
- Queremos seleccionar todos los nodos `<h1>` que sean hijos directos de cualquier nodo `<div>`. Este último puede aparecer en cualquier posición del árbol XML. ¿Cómo lo hacemos?

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos seleccionar todos los nodos `<h1>` que son hijos de un nodo `<body>` que, a su vez, es hijo de un nodo `<html>` que es el nodo raíz del árbol XML. ¿Cómo lo hacemos?

```
/html/body/h1
```

- Queremos seleccionar todos los nodos `<h1>` que aparezcan en cualquier posición del árbol XML. ¿Cómo lo hacemos?

```
//h1
```

- Queremos seleccionar todos los nodos `<h1>` que sean hijos directos de cualquier nodo `<div>`. Este último puede aparecer en cualquier posición del árbol XML. ¿Cómo lo hacemos?

```
//div/h1
```

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos todos los nodos <libro> que aparezcan en el documento en cualquier posición y que tengan un nodo hijo <autor> con el valor "Hunter". ¿Cómo lo hacemos?
- Queremos todos los nodos <libro> que aparezcan en cualquier posición y que tengan un atributo "año" con un valor superior a "1999". ¿Cómo lo hacemos?
- Queremos todos los nodos <libro> que aparezcan en cualquier posición y que tengan un atributo "paginas", independientemente del valor de ese atributo. ¿Cómo lo hacemos?

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos todos los nodos <libro> que aparezcan en el documento en cualquier posición y que tengan un nodo hijo <autor> con el valor "Hunter". ¿Cómo lo hacemos?

```
//libro[autor = "Hunter"]
```

- Queremos todos los nodos <libro> que aparezcan en cualquier posición y que tengan un atributo "año" con un valor superior a "1999". ¿Cómo lo hacemos?

```
//libro[@año > 1999]
```

- Queremos todos los nodos <libro> que aparezcan en cualquier posición y que tengan un atributo "paginas", independientemente del valor de ese atributo. ¿Cómo lo hacemos?

```
//libro[@paginas]
```

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos solo el primero nodo <autor> que encuentre para cada nodo <libro> el cual debe ser un hijo del nodo <bib> que es el nodo raíz. ¿Cómo lo hacemos?
- Queremos solo el primer <autor> de todos. ¿Cómo lo hacemos?

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos solo el primero nodo <autor> que encuentre para cada nodo <libro> el cual debe ser un hijo del nodo <bib> que es el nodo raíz. ¿Cómo lo hacemos?

```
/bib/libro/autor[1]
```

- Queremos solo el primer <autor> de todos. ¿Cómo lo hacemos?

```
//autor[1]
```


2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos todos los nodos <titulo> de los nodos <libro>, pero solo de los libros que tengan un atributo "año" con un valor superior a "1999". ¿Cómo lo hacemos?
- Queremos todos los nodos <titulo> que sean descendientes de <libreria>, pero que no tiene por qué ser hijos directos de <libreria>. Librería es el nodo raíz. ¿Cómo lo hacemos?

2. XPATH: CONCEPTOS ESENCIALES

2.5. Más ejemplos

- Queremos todos los nodos <titulo> de los nodos <libro>, pero solo de los libros que tengan un atributo "año" con un valor superior a "1999". ¿Cómo lo hacemos?

```
//libro[@año>1999]/titulo
```

- Queremos todos los nodos <titulo> que sean descendientes de <libreria>, pero que no tiene por qué ser hijos directos de <libreria>. Librería es el nodo raíz. ¿Cómo lo hacemos?

```
/libreria//titulo
```