|  |
| --- |
|  |
| Localizador XPath |
|  |
| [Escriba el subtítulo del documento] |

**[Escriba el nombre de la compañía]**

[Seleccionar fecha]

Autor: profesor

Localizador XPath

[Escriba el subtítulo del documento]

## Localizador XPath

XPath es un lenguaje capaz de localizar componentes específicos de un documento XML. Ya hemos visto el uso de "/" para seleccionar el nodo raiz y "comics/comic" para seleccionar un elemento dentro de otro.

Crear un sitio WEB que contiene los siguientes elementos:

Fichero XML

<?xml-stylesheet type="text/xsl" href="myfile.xsl" ?>

<bookstore specialty="novel">

<book style="autobiography">

<author>

<first-name>Joe</first-name>

<last-name>Bob</last-name>

<award>Trenton Literary Review Honorable Mention</award>

</author>

<price>12</price>

</book>

<book style="textbook">

<author>

<first-name>Mary</first-name>

<last-name>Bob</last-name>

<publication>

Selected Short Stories of

<first-name>Mary</first-name>

<last-name>Bob</last-name>

</publication>

</author>

<editor>

<first-name>Britney</first-name>

<last-name>Bob</last-name>

</editor>

<price>55</price>

</book>

<magazine style="glossy" frequency="monthly">

<price>2.50</price>

<subscription price="24" per="year"/>

</magazine>

<book style="novel" id="myfave">

<author>

<first-name>Toni</first-name>

<last-name>Bob</last-name>

<degree from="Trenton U">B.A.</degree>

<degree from="Harvard">Ph.D.</degree>

<award>Pulitzer</award>

<publication>Still in Trenton</publication>

<publication>Trenton Forever</publication>

</author>

<price intl="Canada" exchange="0.7">6.50</price>

<excerpt>

<p>It was a dark and stormy night.</p>

<p>

But then all nights in Trenton seem dark and

stormy to someone who has gone through what

<emph>I</emph> have.

</p>

<definition-list>

<term>Trenton</term>

<definition>misery</definition>

</definition-list>

</excerpt>

</book>

<my:book xmlns:my="uri:mynamespace" style="leather" price="29.50">

<my:title>Who's Who in Trenton</my:title>

<my:author>Robert Bob</my:author>

</my:book>

</bookstore>

Una página web que contiene:

Un gridView y un origen de datos XML (XMLDataSource)

<asp:XmlDataSource ID="XmlDataSource1" runat="server" DataFile="~/Ejemplo7/XMLFile.xml" XPath="bookstore/book/author[first-name = 'Toni'] "></asp:XmlDataSource>

<asp:GridView ID="GridView1" runat="server" DataSourceID="XmlDataSource1" AutoGenerateColumns="False">

<Columns>

<asp:TemplateField HeaderText="AUTOR">

<ItemTemplate>

<%# XPath("./first-name")%><br />

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Apellido">

<ItemTemplate>

<%# XPath("./last-name")%><br />

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="awards">

<ItemTemplate>

<%# XPath("./award")%><br />

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

Segundo ejemplo

Introducir este segundo libro en el fichero XML

<book style="novel" id="myfave2">

<author>

<first-name>Toni2</first-name>

<last-name>Bob2</last-name>

<degree from="Trenton U">B.A.</degree>

<degree from="Harvard">Ph.D.</degree>

<award>Pulitzer</award>

<publication>Still in Trenton</publication>

<publication>Trenton Forever</publication>

</author>

<price intl="Canada" exchange="0.7">7.50</price>

<excerpt>

<p>It was a dark and stormy night.</p>

<p>

But then all nights in Trenton seem dark and

stormy to someone who has gone through what

<emph>I</emph> have.

</p>

<definition-list>

<term>Trenton</term>

<definition>misery</definition>

</definition-list>

</excerpt>

</book>

Crear una nueva página asp con el siguiente contenido

<asp:XmlDataSource

id="XmlSource"

DataFile="~/Ejemplo7/XMLFile.xml"

runat="server"

XPath="bookstore/book[@style='novel']"/>

<asp:GridView ID="GridView2" runat="server" DataSourceID="XmlDataSource" AutoGenerateColumns="False">

<Columns>

<asp:TemplateField HeaderText="AUTOR">

<ItemTemplate>

<%# XPath("./first-name")%><br />

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="Apellido">

<ItemTemplate>

<%# XPath("./last-name")%><br />

</ItemTemplate>

</asp:TemplateField>

<asp:TemplateField HeaderText="awards">

<ItemTemplate>

<%# XPath("./award")%><br />

</ItemTemplate>

</asp:TemplateField>

</Columns>

</asp:GridView>

<asp:Repeater ID="Repeater1"

DataSourceID="XmlSource"

runat="server">

<ItemTemplate>

<b>Price:</b>

<%# XPath("./price")%>

</ItemTemplate>

</asp:Repeater>

# ++Información Sobre XPath

//Si queremos seleccionar un elemento, sin importar en que otros elementos esta anidado, usaremos la doble barra invertida. Por ejemplo "//titulo" localiza todos los elementos título de un documento

atributos: @

La arroba sirve para localizar atributos: "@lang" por ejemplo indica el atributo lang.

nodos hijos: \*

El carácter \* localiza todos los nodos hijos, **sin incluir descendientes**

todos los nodos : node()

La función node( ) localiza todos los nodos descendientes, incluidos hijos

nodo actual: .

El punto selecciona el nodo actual.

Seleccionar ascendiente: ..

Dos puntos seleccionan el nodo ascendiente del actual.

### Funciones XPath

XPath define ciertas funciones como por ejemplo:

position()

Devuelve la posición de un elemento.

last()

Devuelve la posición del último elemento.

name()

Devuelve el nombre de la etiqueta.

namespace-uri()

Devuelve el nombre de espacio del elemento actual

### Funciones de cadena

string(selector)

Convierte a cadena. Si el selector devuele un conjunto de nodos (nodeset) devuelve el primero.

starts-with(pajar,aguja)

Si la cadena parar comienza con aguja.

contains(pajar,aguja)

substring-before(pajar, aguja)

substring-after(pajar, aguja)

substring (cadena, inicio [,longitud])

string-length(cadena)

normalize-space(cadena)

Elimina espacios extra

string(true()) ->“true” starts-with(“Pepelui”, “Pe”) ->true() contains(“aeIou”, “eI”) -> true() substring-before(“DD/MM/AAAA”, “/”) -> “DD” substring-after(“DD/MM/AAAA”, “/”) -> “MM/AAAA” substring(“DD/MM/AAAA”, 3) -> “/MM/AAAA” substring(“DD/MM/AAAA”, 3) -> “/MM/AAAA” normalize-space(cadena) -> Elimina espacios extra

### Funciones de booleanas

Los desarrolladores de XPath añadierón una serie de funciones booleanas, con el proposito de construir una espeficiación estricta.

boolean()

Convertir un valor en booleano

true()

Es la representación XPATH del booleano true.

false()

Representación XPATH del booleano false.

not(expresion)

Niega la expresión.

### Funciones de agregados

XPath cuenta com dos estas funciones de agregados:

count ( selector-XPath )

Devuelve el número de elemento, generados por el selector

sum ( selector-XPath )

Suma el contenido de los elemento, generados por el selector

**Contar el numer de comics** <xsl:value-of select="count(comics/comic)" /> **Contar el numer de comics en ingles** <xsl:value-of select="count(comics/comic[@lang='en')" /> **Sumar el precio de los comics en castellano** <xsl:value-of select="sumt(comics/comic[@lang='es']/precio" />

### Predicados

Un predicado permite restringir el conjunto de nodos devueltos por un XPath. Se encierran siempre entre corchetes, dentros de los cuales podemos poner:

un número: comics/comic[2]

Devuelve el segundo comic.

la función last: comics/comic[last()]

Devuelve el ultimo comic.

una evaluacion: comics/comic[@lang='fr']

Devuelve todos los comic cuyo atributo "lang" sea igual a "fr".

una evaluacion: comics/comic[@precio &lt; 10]

Todos los comics con precio superior a 10.