Tema 10

Gestión y recuperación de información. XPath

Objetivos

- Utilizar los documentos XML como método para el almacenamiento de la información.
- Identificar el ámbito de aplicación del almacenamiento de información en formato XML.
- Comprender las ventajas y los inconvenientes de almacenar información en formato XML.
- Identificar las características de los sistemas de almacenamiento de información.
- Utilizar las técnicas de búsqueda de información en documentos XML.
- Conocer y utilizar el lenguaje XML de consulta de información XPath *(XML Path)*.
- Utilizar herramientas de consulta de información en formato XML.

Introducción

En los temas anteriores se han estudiado las características generales de los documentos XML en relación con que estuvieran bien formados y fueran válidos. Una de las características principales de los documentos XML es su capacidad para almacenar información estructurada en formato de texto. En este tema y en el siguiente se estudian algunas de las tecnologías XML que permiten recuperar la información almacenada en formato XML, utilizando para ello lenguajes de consulta específicos. En este tema se aborda el estudio del lenguaje de consulta XPath *(XML Path)* y en el tema siguiente se estudiará el lenguaje de consulta XQuery (XML Query). El uso del lenguaje de consulta XPath *(XML Path)* es sencillo y su importancia radica en que suele emplearse en combinación con otras tecnologías XML, como, por ejemplo, la tecnología XSLT *(eXtensible Stylesheet Language Transformations)*.

Índice

10.1 ALMACENAMIENTO Y CONSULTA DE INFORMACIÓN EN DOCUMENTOS XML

La utilización del lenguaje de marcas XML, y sus tecnologías asociadas, permiten establecer un sistema de almacenamiento de información fiable y ordenado. Se puede considerar que los documentos XML que tengan una misma estructura, tendrán también un contenido de información homogéneo, es decir, que será semejante. De modo que la estructura de los documentos XML es la que garantiza la compatibilidad de la información almacenada. Por este motivo, precisamente, es necesario disponer de las técnicas y herramientas que permitan validar los documentos XML creados, mediante declaraciones DTD u otras técnicas de validación, para poder comprobar la adaptación del contenido de información a la estructura prestablecida. Una vez que un documento XML se ha validado, y por tanto su estructura es la adecuada, entonces se dispone de un sistema de almacenamiento de información en formato XML con la capacidad de poder extraer esa información del documento mediante la utilización de un lenguaje de consulta XML.

El almacenamiento de información es, actualmente, una parte fundamental de la gestión y administración de las organizaciones. El uso del lenguaje XML está permitiendo romper barreras y crear nuevas formas de almacenar y administrar información. Estas nuevas formas, sustancialmente, se apartan del uso tradicional de bases de datos relacionales basadas en almacenamiento en formato tabular (tablas, filas o registros y columnas o campos).

Actualmente, los sistemas de almacenamiento pueden ser clasificados en dos categorías: los centrados en los datos y los centrados en los documentos. Los sistemas de almacenamiento centrados en datos son altamente estructurados, con tipos de datos de tamaño limitado, reglas poco flexibles para campos opcionales y contenido basado en registros. Los sistemas de almacenamiento centrados en documentos tienden a ser más impredecibles en tamaño y en contenido. Las bases de datos relacionales suelen ser mejores para tratar con los requerimientos de almacenamiento centrados en los datos, mientras que los sistemas de almacenamiento basados en formato XML son típicamente mejores para almacenar datos centrados en los documentos. En la actualidad, existen sistemas gestores de bases de datos específicos para el tratamiento y la recuperación de la información en documentos XML, como por ejemplo BaseX. Estos sistemas gestores de bases de datos específicos que soportan almacenamiento en formato XML suelen recibir el nombre de Bases de datos XML nativas o NXD *(Native XML Database)*. Las fronteras entre las dos categorías de sistemas almacenamiento nunca han quedado del todo claras. Tanto es así que, actualmente, las bases de datos relacionales suelen también ofrecer soporte y herramientas para el tratamiento y consulta de información almacenada en formato XML.

La utilización de documentos XML como sistema de almacenamiento no se queda solo en la capacidad de almacenar información, sino también en la capacidad para extraer la información. Así, si un documento está bien formado y es válido, será posible localizar sus elementos y la información que contienen éstos, de forma sencilla. Efectivamente, si un documento XML tiene una estructura bien definida es posible acceder a la información contenida para, posteriormente, poder tratarla de la manera más adecuada a través de los procesamientos necesarios. El consorcio W3C fue consciente de todo ello y publicó las especificaciones de los dos lenguajes de consulta de información sobre documentos XML: XPath *(XML Path)* y XQuery *(XML Query)*.

- XPath. Es un lenguaje básico y sencillo de expresiones de consulta que permite acceder a partes de un documento XML.

- XQuery. Es otro lenguaje de consultas más complejo que permite la manipulación de la información almacenada en documentos XML. El lenguaje XML de consulta XQuery hace uso del lenguaje de consulta XPath.

10.2 ESTRUCTURA JERÁRQUICA EN ÁRBOL

Para comprender el sistema de almacenamiento de información en documentos XML y, por tanto, el funcionamiento de los lenguajes de consulta, tanto con XPath como con Xquery, es necesario considerar un documento XML como un árbol de nodos. Esta forma de ver los documentos XML da lugar a lo que se conoce como estructura jerárquica en árbol de los documentos XML.

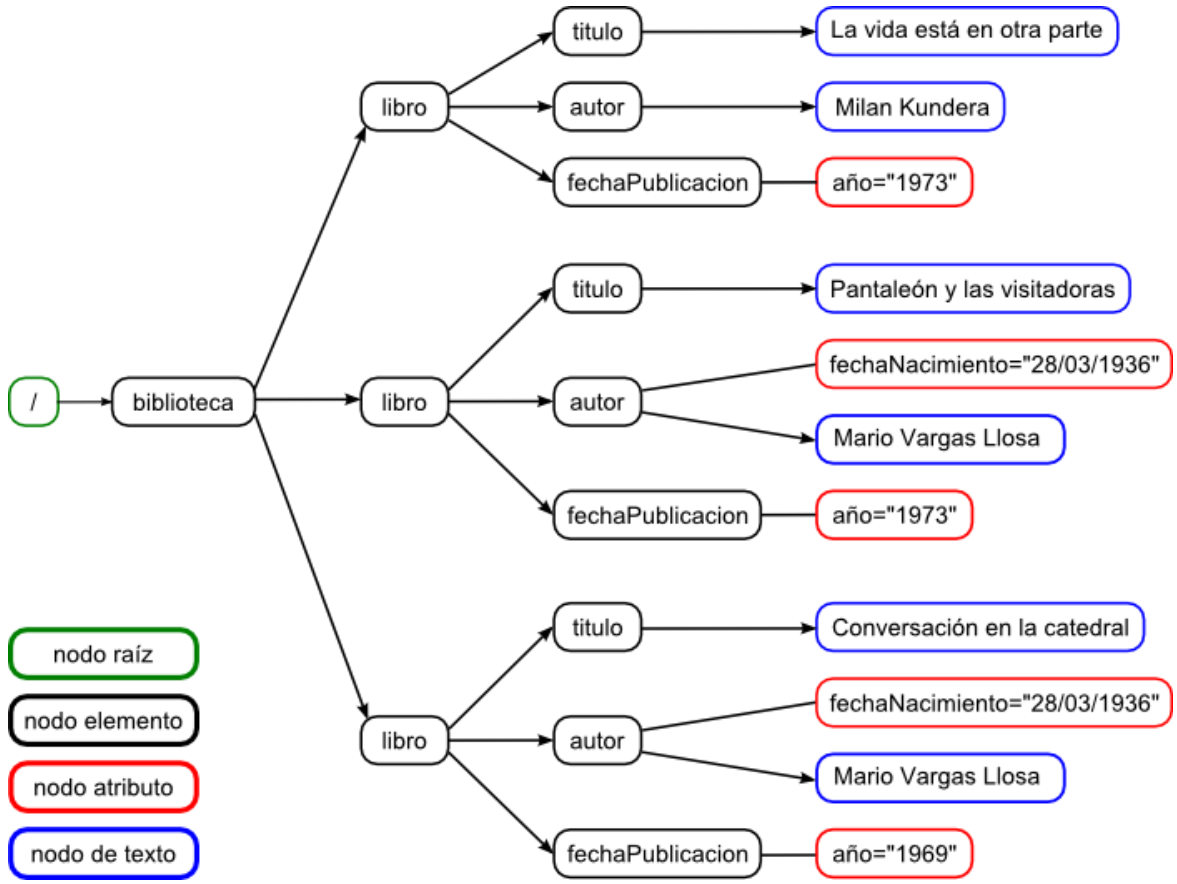
Un documento XML puede representarse como un **árbol dirigido**, para expresar la estructura jerárquica existente entre sus elementos o componentes. Así, se puede considerar, por ejemplo, que los elementos son nodos y que uno de los elementos que lo forman es el elemento padre que contiene al resto de elementos. Pero en los lenguajes de consulta de documentos XML no sólo los elementos son nodos, en realidad hay siete tipos de nodos:

- Raíz.
- Elemento.
- Atributo.
- Texto o información textual contenida en un elemento.
- Comentario.
- Instrucción de procesamiento.
- Espacio de nombres.

Por ejemplo, el siguiente documento XML:

```
<?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>
```

se puede representar gráficamente mediante un árbol dirigido que puede apreciarse en la siguiente ilustración. Puede apreciarse que el nodo raíz (verde), que los nodos elemento (negro), que los nodos atributo (rojo) y que los nodos de texto (azul) están diferenciados entre sí. Debe tenerse en cuenta que los nodos atributo y de texto no pueden tener nodos descendientes. En realidad, los nodos atributo y de texto se consideran como hijos del nodo que representa el elemento correspondiente. Las expresiones de consulta XPath suelen actuar sobre los nodos elemento, sin embargo, para referirse a los atributos o al texto se utilizan notaciones especiales.



No existe una norma sobre la forma de representar el contenido del árbol. En este caso de ejemplo se ha reflejado el nombre de los elementos y su valor contenido dentro de él. De esta forma, puede apreciarse, claramente, la estructura jerárquica de los elementos, así como, los valores de los atributos y del texto de los elementos que, además, no tienen nodos descendientes.

10.3 LENGUAJE DE CONSULTA XPATH

En el apartado anterior, se ha estudiado la estructura jerárquica en forma de árbol que posee un documento XML. Así, en un documento XML hay elementos que contienen a otros elementos y otros que contienen información textual y que pueden contener, además, atributos.

Considerando la estructura jerárquica en árbol de los documentos XML, se usa XPath para recorrer los elementos del documento y extraer la información contenida. XPath es un estándar de W3C que define un conjunto de reglas para referenciar partes de un documento XML, de manera que se trata de un lenguaje de expresiones de consulta que se usa para buscar y encontrar información en documentos XML. Hay que tener en cuenta que XPath se utiliza siempre en el contexto de un lenguaje anfitrión, por ejemplo, con el lenguaje XSLT *(eXtensible Stylesheet Language Transformations)* que permite realizar transformaciones sobre documentos XML. De manera que XPath no se utiliza de modo independiente, sino de forma conjunta con otras tecnologías.

Las características más significativas del lenguaje XPath son las siguientes:

- Proporciona una sintaxis específica para referenciar partes de un documento XML.
- Utiliza expresiones que referencian las rutas, o los caminos o *paths* a recorrer hasta alcanzar un determinado elemento dentro de la estructura jerárquica en árbol de un documento XML. Las búsquedas de la información contenida en los elementos XML se definen mediante expresiones que hacen referencia a estas rutas.
- Contiene una librería de funciones.
- Es un estándar de W3C.

En resumen, XPath es un lenguaje de expresiones de consulta que trata un documento XML como un árbol de nodos. De manera que utiliza la definición de expresiones de búsqueda de tipo ruta para recorrer el árbol, encontrar y extraer la información correspondiente del documento XML.

Para abordar el estudio de XPath se utiliza el siguiente documento XML de ejemplo.

```
<?xml version="1.0" encoding="utf-8" ?>
<libreria>
  <libro categoria="Infantil">
    <titulo idioma="inglés">Harry Potter</titulo>
    <autor>J.K. Rowling</autor>
    <anyo>2005</anyo>
    <precio>19.99</precio>
  </libro>
  <libro categoria="Informática">
    <titulo idioma="español">Aprendiendo XML</titulo>
    <autor>Erik T. Ray</autor>
    <anyo>2003</anyo>
    <precio>39.95</precio>
  </libro>
  <libro categoria="Cocina">
    <titulo idioma="inglés">Everyday Italian</titulo>
    <autor>Giada De Laurentiis</autor>
    <anyo>2005</anyo>
    <precio>30.00</precio>
  </libro>
</libreria>
```

En el ejemplo anterior, puede observarse que el elemento raíz se denomina libreria. El elemento raíz incluye tres elementos libro que, a su vez, incluyen elementos que representan características del libro: título, autor, año de publicación y precio. Estos elementos contienen la información textual correspondiente. Además, puede apreciarse que el elemento libro tiene un nodo atributo, denominado categoria y el elemento título tiene un nodo atributo, denominado idioma.

También, en el ejemplo anterior pueden apreciarse las **relaciones jerárquicas** entre los elementos:

- **Padres.** Cada nodo de elemento o atributo tiene un nodo padre. Por ejemplo, el nodo libro es el nodo padre de los nodos titulo, autor, anyo y precio. Y, el nodo titulo es el nodo padre del atributo idioma.

- **Hijos:** Todo elemento tiene 0, 1 o más hijos. Así, los nodos de los elementos titulo, autor, anyo y precio son nodos hijo del elemento libro.

- **Hermanos.** Son los nodos que tienen el mismo padre. Por ejemplo, titulo, autor, anyo y precio son nodos hermanos.

- **Ancestros.** Se denominan nodos ancestros todos los nodos superiores dentro de la estructura jerárquica del documento, es decir, el padre de un nodo, el padre del padre, etc. Por ejemplo, los nodos libro y librería son los nodos ancestros del nodo titulo.

- **Descendientes.** Se denominan nodos descendientes todos los nodos inferiores dentro de la estructura jerárquica del documento, es decir, el hijo de un nodo, el hijo del hijo, etc. Por ejemplo, los nodos libro, titulo, autor, anyo y precio son nodos descendientes del nodo librería.

Selección de nodos

Las expresiones XPath de selección de nodos, que también reciben el nombre de expresiones de direccionamiento, pueden ser absolutas o relativas. Las **expresiones XPath absolutas** son aquellas que empiezan por el nodo raíz, mientras que las **expresiones XPath relativas** son aquellas que se construyen con respecto a un nodo determinado, denominado nodo de contexto o nodo actual. La siguiente tabla muestra las expresiones XPath más básicas que pueden utilizarse para seleccionar nodos.

Selección	Descripción
nombre_nodo	Selecciona todos los nodos denominados nombre_nodo.
/	Selecciona desde el nodo raíz. Nodos hijos directos
//	Selecciona los nodos del documento desde el nodo actual que coinciden con la selección, no importa dónde se encuentren. Nodos descendientes
.	Selecciona el nodo actual.
..	Selecciona el padre del nodo actual.
@	Selecciona atributos.

En la siguiente tabla pueden observarse diversos ejemplos de expresiones XPath sobre el documento XML de ejemplo.

Expresión XPath	Resultado
libreria	Selecciona todos los nodos con el nombre libreria.
/libreria	Selecciona el nodo raíz libreria.
/libreria/libro	Selecciona todos los elementos libro que son hijos del nodo libreria.
libreria/libro	Selecciona todos los elementos libro que son hijos del nodo libreria.
//libro	Selecciona todos los elementos libro no importa donde se encuentren en el documento.
libreria//titulo	Selecciona todos los elementos titulo que son descendientes del elemento libreria, sin importar donde se encuentren debajo del elemento libreria.
//titulo/..	Selecciona todos los elementos libro que son padre del elemento título.
//@idioma	Selecciona todos los atributos del documento denominados idioma.

Selección de nodos desconocidos. Comodines

Los comodines de XPath se usan para seleccionar elementos desconocidos.

Comodín	Descripción
*****	Representa cualquier elemento nodo.
@*	Representa cualquier atributo nodo.
node()	Representa cualquier nodo de cualquier clase.

La siguiente tabla muestra la aplicación de diversos ejemplos sobre el documento XML de ejemplo.

Expresión XPath	Resultado
/libreria/*	Selecciona todos los nodos hijo del elemento libreria.
/libreria/libro/*	Selecciona todos los elementos hijo de todos los elementos libro del elemento libreria.
/libreria/*/titulo	Selecciona todos los elementos titulo que son nietos del elemento librería, independientemente del elemento padre.
//*	Selecciona todos los elementos del documento.
//titulo[@*]	Selecciona todos los elementos titulo que tengan algún atributo.

Predicados

Los predicados se utilizan para encontrar un nodo específico, o bien, un nodo que contiene un valor específico. Los predicados están siempre dentro de corchetes. En la siguiente tabla se pueden observar diversos ejemplos de aplicación sobre el documento XML de ejemplo.

Expresión XPath	Resultado
/libreria/libro[1]	Selecciona el primer elemento libro que es hijo del elemento libreria.
/libreria/libro[last()]	Selecciona el último elemento libro que es hijo del elemento libreria.
/libreria/libro[last()-1]	Selecciona el penúltimo elemento libro que es hijo del nodo libreria.
/libreria/libro[position()<3]	Selecciona los primeros dos elementos libro que son hijos del elemento libreria.
/libreria/libro[precio]	Selecciona todos los elementos libro que son hijos del elemento libreria y que contengan un elemento precio.
/libreria/libro[anyo=2003]	Selecciona todos los elementos libro que son hijos del elemento libreria y que tienen un elemento anoy con un valor igual a 2003.
/libreria/libro[precio>35.00]	Selecciona todos los elementos libro que son hijos del elemento libreria y que tienen un elemento precio con un valor mayor de 35.00.
/libreria/libro[precio>35.00]/titulo	Selecciona todos los elementos titulo que son hijos del elemento libro y que tienen un elemento precio con un valor mayor de 35.00.
//libro[autor="J.K. Rowling"]	Selecciona todos los libros cuyo autor es “J.K. Rowling”
//titulo[@idioma]	Selecciona todos los elementos titulo que tienen un atributo llamado idioma.
//titulo[@idioma='inglés']	Selecciona todos los elementos titulo que tienen un atributo llamado idioma y cuyo valor es inglés.

Cabe destacar que no existe una función first() en contraposición a last(), por lo que para acceder al primer elemento en su lugar debe utilizarse el marcador 1 como valor del predicado.

Selección de varias rutas

Se puede utilizar el operador | para obtener la unión de resultados de búsqueda. Se suele emplear para seleccionar varias rutas. En la siguiente tabla se muestran diversos ejemplos de uso:

Expresión XPath	Resultado
//libro/titulo | //libro/precio	Selecciona todos los elementos titulo y *(AND)* todos los elementos precio de todos los elementos libro.
//titulo | //precio	Selecciona todos los elementos titulo y *(AND)* todos los elementos precio del documento.
/libreria/libro/titulo | //precio	Selecciona todos los elementos titulo de los elementos libro del elemento librería y todos los elementos precio del documento.

Ejes XPath

Un Eje XPath define un conjunto de nodos en relación con el nodo actual. Son los siguientes:

Nombre del Eje	Descripción
ancestor	Selecciona todos los ancestros (padre, abuelo, etc.) del nodo actual.
ancestor-or-self	Selecciona todos los ancestros (padre, abuelo, etc.) del nodo actual, incluyendo el nodo actual.
attribute	Selecciona todos los atributos del nodo actual.
child	Selecciona todos los hijos del nodo actual. Es el eje por defecto.
descendant	Selecciona todos los descendientes (hijo, nieto, etc.) del nodo actual.
descendant-or-self	Selecciona todos los descendientes (hijo, nieto, etc.) del nodo actual, incluyendo el nodo actual.
following	Selecciona todo aquello hay en el documento después de la etiqueta de cierre del nodo actual.
following-sibling	Selecciona todos los hermanos que hay después del nodo actual.
namespace	Selecciona todos los nodos namespace del nodo actual.
parent	Selecciona el padre del nodo actual.
preceding	Selecciona todos los nodos que hay antes del nodo actual, excepto los ancestros, los atributo y los nodos namespace.
preceding-sibling	Selecciona todos los hermanos que hay antes del nodo actual.
self	Selecciona el nodo actual.

En una expresión XPath, una ruta de localización de nodos consiste en uno o más nombres de nodos separados por caracteres /. Para realizarse una búsqueda XPath, se evalúa paso a paso cada uno de los nodos de la ruta sobre el conjunto actual de nodos. En cada uno de estos pasos se evalúa:

- El eje XPath *(axisname)* que define la relación en el árbol, entre los nodos seleccionados y el nodo actual. El eje child:: está establecido por defecto, por lo que se suele omitir.
- El test de nodo *(nodetest)* que identifica un nodo dentro de un eje XPath.
- Los predicados *(predicate)* existentes en la expresión para afinar la definición del conjunto de nodos seleccionado.

La sintaxis para un paso de localización de nodos es la siguiente: **axisname::nodetest[predicate]**. En XPath existen dos formas de sintaxis: la completa y la abreviada. La especificación de ejes da lugar a la sintaxis completa. Por ejemplo, las siguientes expresiones XPath son equivalentes:

Sintaxis completa: /child::libreria/child::libro/child::titulo/attribute::idioma
Sintaxis abreviada: /libreria/libro/titulo/@idioma

Se recomienda utilizar siempre la sintaxis abreviada para formar expresiones de consulta XPath, porque es el tipo de sintaxis que se suele emplear habitualmente. Sin embargo, por curiosidad, la siguiente tabla muestra diversos ejemplos de sintaxis completa sobre el documento XML de ejemplo.

Expresión XPath	Resultado
/libreria/child::libro	Selecciona todos los nodos libro que son hijos del nodo libreria.
//titulo/attribute::idioma	Selecciona el atributo idioma de los nodos titulo.
//libro/child::*	Selecciona todos los elementos que son hijos del nodo libro.
//titulo/attribute::*	Selecciona todos los atributos del nodo libro.
//libro/child::node()	Selecciona todos los elementos que son hijos del nodo libro.
/libreria/descendant::libro	Selecciona todos los nodos libro que son descendientes del nodo libreria.
//titulo/ancestor::libro	Selecciona todos los nodos libro que son ancestros del nodo titulo.
//titulo/ancestor-or-self::libro	Selecciona todos los libros ancestros del nodo titulo y el nodo actual también si es un nodo libro.
/libreria/child::*/child::precio	Selecciona todos los precios que son nietos del nodo libreria.

Operadores XPath

Una expresión XPATH devuelve un conjunto de nodos, un string, un boolean o un número. Para obtener el resultado deseado en cada caso, pueden emplearse los siguientes operadores formando parte de las expresiones XPath.

Operador	Descripción	Ejemplo
|	AND. Unión de resultados.	//libro | //cd
+	Suma.	6 + 4
-	Resta.	6 - 4
*****	Multiplicación.	6 * 4
div	División.	8 div 4
=	Igualdad.	precio=9.80
!=	No igual (distinto).	precio!=9.80
<	Menor que.	precio<9.80
<=	Menor o igual que.	precio <=9.80
>	Mayor que.	precio >9.80
>=	Mayor o igual que.	precio >=9.80
or	OR	precio =9.80 or precio =9.70
and	AND	precio >9.00 and precio <9.90
mod	Modulo, resto de la división entera.	5 mod 2

Funciones XPath

El lenguaje XPath proporciona una librería de funciones que pueden utilizarse formando parte de los predicados para afinar o mejorar la selección. Por ejemplo, anteriormente, ya se han utilizado la función last(). La siguiente tabla presenta algunas de las funciones disponibles con XPath.

Función	Sintaxis	Descripción
count()	count(nodos)	Devuelve el número de nodos que pertenecen al conjunto de nodos proporcionado.
id()	id(valor)	Selecciona nodos por su ID único.
last()	last(nodos)	Devuelve la posición del último nodo del conjunto de nodos.
name()	name()	Devuelve una cadena con el nombre del nodo actual.
position()	position(n)	Devuelve el elemento que se encuentra en la posición n.
substring()	substring(valor,a,b)	Devuelve una subcadena del valor entre la posición a y b.
translate()	translate(cad,c1,c2)	En la cadena cad, sustituye la subcadena c1 por c2.
concat()	concat(c1,c2)	Devuelve una cadena que representa la concatenación de la cadena c1 con la cadena c2.
contains()	conains(c1,c2)	Devuelve verdadero *(true)* si c1 contiene c2.
string()	string()	Convierte un objeto en cadena.
false()	false()	Devuelve *true*.
true()	true()	Devuelve *false*.
not()	not(argumento)	Devuelve *true* si el argumento es *false*, y viceversa.
sum()	sum(nodos)	Devuelve la suma de valores del conjunto de nodos.
round()	round(argumento)	Devuelve un entero con el valor más próximo al argumento.
number()	number(argumento)	Devuelve el número correspondiente al argumento.
ceiling	ceiling(decimal)	Evalúa un número decimal y devuelve el entero más pequeño mayor o igual que ese número decimal.
floor	floor(decimal)	Evalúa un número decimal y devuelve el número entero más grande menor o igual que ese número decimal.
max()	max(nodos)	Devuelve el valor máximo del conjunto de nodos
min()	min(nodos)	Devuelve el valor mínimo del conjunto de nodos
node()	node()	Representa cualquier nodo de cualquier clase.
text()	text()	Devuelve el contenido textual de un elemento.
data()	data()	Devuelve el contenido textual de un atributo

Acceso al contenido textual de elementos y atributos

Cuando se evalúa una ruta de localización de XPath se devuelve un conjunto de nodos. Por ejemplo, al evaluarse la expresión: /libreria/libro/titulo, se obtienen todos los elementos descendientes de los nodos titulo. Es decir, se obtiene el siguiente resultado:

```
<titulo idioma="ingles">Harry Potter</titulo>
<titulo idioma="español">Aprendiendo XML</titulo>
<titulo idioma="ingles">Everyday Italian</titulo>
```

Para acceder al **valor del texto contenido en un nodo o lista de nodos** resultante de una expresión XPath, se emplea la función **text()** de XPath.

Así, por ejemplo, si se evalúa la expresión:

/libreria/libro/titulo/text()

se obtendrá el siguiente resultado:

```
Harry Potter
Aprendiendo XML
Everyday Italian
```

Para acceder al **valor del texto contenido en los atributos** de los elementos se emplea la función **data()** de XPath. Hay que tener en cuenta que, para seleccionar un atributo de un elemento se debe incluir el carácter **@** delante del nombre del atributo, evidentemente, una vez que ha sido definida adecuadamente la ruta de localización del atributo correspondiente.

Así, por ejemplo, si se evalúa la expresión:

/libreria/libro/titulo/@idioma/data()

se obtendrá el siguiente resultado:

```
ingles
español
ingles
```

Cuando un analizador de XPath evalúa una expresión de consulta produce como resultado un texto xml que corresponde con la respuesta de la expresión XPath evaluada. En caso de que no existiera la ubicación establecida en la ruta de la expresión XPath, se devolvería el conjunto vacío.

Los analizadores de XPath pueden actuar de manera diferente a la hora de presentar los resultados del contenido textual de elementos y atributos, por este motivo se utilzan las funciones text() y data() cuando se desea acceder al contenido de texto incluido en los elementos y los atributos, respectivamente. Habitualmente, los analizadores de expresiones XPath pueden devolver directamente el contenido textual del nodo o nodos localizados, o también, pueden devolver el contenido textual de los atributos de estos nodos. También hay que tener en cuenta que existen analizadores de XPath que no incorporan esta funcionalidad.

A continuación, se incluyen algunos ejemplos de expresiones de consulta resueltos que se pueden ejecutar sobre el documento XML de ejemplo. La siguiente tabla muestra los resultados producidos al evaluar las expresiones XPath de ejemplo.

Expresiones XPath del ejemplo	Resultado
/libreria/libro/titulo	<titulo idioma="inglés">Harry Potter</titulo> <titulo idioma="español">Aprendiendo XML</titulo> <titulo idioma="inglés">Everyday Italian</titulo>
/libreria/libro[1]/titulo	<titulo idioma="inglés">Harry Potter</titulo>
/libreria/libro/precio	<precio>19.99</precio> <precio>39.95</precio> <precio>30.00</precio>
/libreria/libro/precio/text()	19.99 39.95 30.00
/libreria/libro/precio[text()]	<precio>19.99</precio> <precio>39.95</precio> <precio>30.00</precio>
/libreria/libro[precio>35]/precio	<precio>39.95</precio>
/libreria/libro[precio>35]/titulo	<titulo idioma="español">Aprendiendo XML</titulo>
/libreria/libro[@categoria="Informática"]/titulo	<titulo idioma="español">Aprendiendo XML</titulo>
//titulo[@idioma="inglés"]/text()	Harry Potter Everyday Italian
/libreria/libro[3]/autor/text()	Giada De Laurentiis
//libro[titulo="Aprendiendo XML"]/autor	<autor>Erik T. Ray</autor>
//libro/titulo[contains(text(),"Har")]	<titulo idioma="inglés">Harry Potter</titulo>
//libro[titulo="Harry Potter"]/@categoria/data()	Infantil
//libro[anyo="2005"]/@categoria/data()	Infantil Cocina

10.4 HERRAMIENTAS PARA LA CONSULTA DE INFORMACIÓN EN FORMATO XML

Como se ha podido comprobar a lo largo del tema, XPath es un lenguaje de consulta que se utiliza para seleccionar elementos y atributos de un documento XML navegando a través de su estructura en forma de árbol. Se utilizan expresiones de consulta Xpath para seleccionar nodos o conjuntos de nodos del documento XML en función de determinados criterios de búsqueda.

Existen diversos tipos de herramientas informáticas que incluyen un analizador de XPath. En la actualidad, los editores de código XML específicos suelen incluir un analizador de XPath que permite procesar expresiones de consulta XPath. Además, las herramientas de desarrollo XML incluyen funciones de procesamiento XPath. Algunos de los editores de código y herramientas de desarrollo XML más extendidos que incluyen XPath son los siguientes: XML Copy Editor, Jaxe XML Editor, XMLSpy, Oxigen XML, Stylus Studio, Visual Studio, etc.

Por otra parte, existen herramientas informáticas específicas que facilitan la extracción de información en documentos XML utilizando expresiones de consulta XPath. En general, estas herramientas permiten extraer información almacenada en documentos XML a partir de expresiones de consulta XPath y XQuery. Algunas de estas herramientas específicas que suelen considerarse como bases de datos XML nativas son: BaseX, eXist, DBDOM, Berkeley DB XML y dbXML.

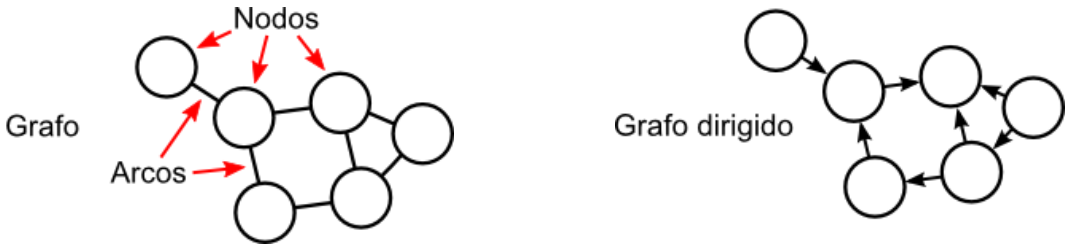
Actualmente, también se pueden utilizar herramientas on-line para procesar expresiones de consulta XPath sobre documentos XML. Algunas de las cuales son las siguientes:

- Videlibri: http://videlibri.sourceforge.net/cgi-bin/xidelcgi
- XPath Tester: http://www.xpathtester.com/xpath
- FreeFormatter: https://www.freeformatter.com/xpath-tester.html
- Code Beauty: https://codebeautify.org/Xpath-Tester

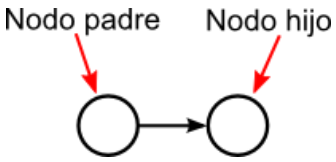
Glosario de términos

Árbol de nodos. Un árbol es una estructura de datos que equivale a un árbol. A su vez, un árbol es un caso particular de grafo. Los grafos permiten estudiar las interrelaciones existentes entre los elementos que interactúan entre sí unas con otras dentro de un conjunto. Estos conceptos están definidos en la teoría de grafos. El primer artículo científico relativo a grafos fue escrito por el matemático suizo Leonhard Euler en 1736. Los grafos son ampliamente utilizados en la disciplina informática y su aplicación es particularmente útil cuando se emplean junto con los lenguajes de consulta, como XPath, para acceder a la información estructurada de forma jerárquica, que está contenida en los documentos XML.

Un **grafo** es un conjunto de objetos llamados nodos o vértices unidos por enlaces llamados arcos o aristas. Un **grafo dirigido** es un grafo en el que los arcos tienen dirección.

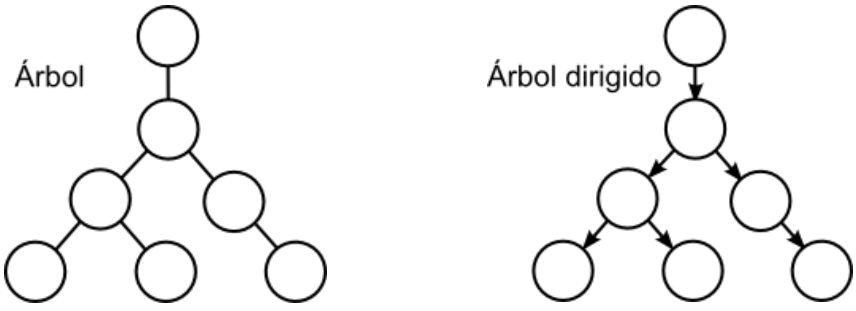


Cuando dos nodos están unidos por un arco con dirección, el **nodo padre** es el nodo del que parte el arco y el **nodo hijo** es el nodo al que llega el arco.

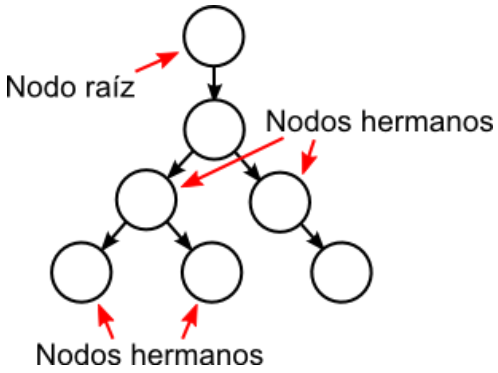


Un **árbol** es un grafo en el que cualquier pareja de vértices están conectados por un único camino, es decir, que no hay ciclos.

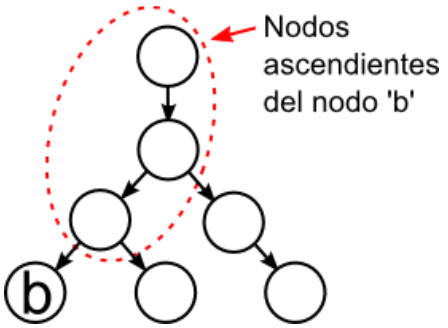
Árbol dirigido. Es un árbol en el que las aristas tienen dirección y todos los nodos menos uno, tienen un único padre.

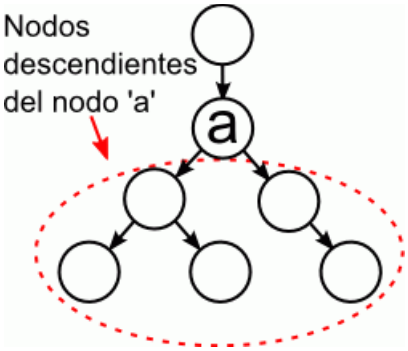


El **nodo raíz** de un árbol dirigido es el único nodo sin padre. Los **nodos hermanos** son los nodos que tienen el mismo padre.



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





Grafo. Es un conjunto de objetos llamados nodos o vértices unidos por enlaces llamados arcos o aristas.