



Σavante

12

LENGUAJES DE MARCAS Y SISTEMA DE GESTIÓN DE INFORMACIÓN

Conversión y adaptación de documentos XML

ÍNDICE

/ 1. Introducción y contextualización práctica	3
/ 2. Lenguaje de hojas extensibles (XSL): XPath	4
/ 3. El árbol: Nodos ascendientes y descendientes	4
/ 4. Sintaxis XPath. Eje	5
/ 5. Caso práctico 1: “Instituto de FP: Árbol”	6
/ 6. Sintaxis XPath. Predicado	7
/ 7. Sintaxis XPath. Selección de nodos	8
/ 8. Sintaxis XPath. Expresiones anidadas	9
/ 9. Caso práctico 2: “Instituto de FP: Consultas”	10
/ 10. Herramientas de validación de expresiones	10
/ 11. Resumen y resolución del caso práctico de la unidad	12
/ 12. Bibliografía	13
12.1. Webgrafía	13

OBJETIVOS

Conocer lenguajes de transformación.

Saber recorrer un árbol XML.

Dominar las expresiones XPath.

/ 1. Introducción y contextualización práctica

Un fichero XML contiene información estructurada y sería muy interesante poder transformar esos datos para obtener nuevos documentos o informes adaptados. Hasta ahora, estamos trabajando con pequeños documentos XML, pero se ha de tener en cuenta que en muchas ocasiones la información va a ser muy extensa y se necesitan mecanismos de filtrado.

Una de las tecnologías actuales y que está creciendo exponencialmente es IoT (Internet of Things o Internet de las cosas), que trabaja con miles de datos y como intuimos, se debe apoyar en mecanismos como los que estudiaremos en este tema para que se consiga filtrar esa información.

Alberto necesita analizar un fichero con datos meteorológicos porque tienen que diseñar una web sobre la previsión del tiempo.

Escucha el siguiente audio donde planteamos la contextualización práctica de este tema, encontrarás su resolución en el apartado Resumen y resolución del caso práctico.



Fig. 1. Personajes.



Audio intro. "Inventario de ordenadores"

<https://bit.ly/31HgeoF>



/ 2. Lenguaje de hojas extensibles (XSL): XPath

En temas anteriores, se ha visto que, gracias al CSS, podemos aplicar estilos al documento HTML y transformarlo.

Las hojas de estilo están pensadas para una transformación visual, pero a nivel de transformación de información, está muy limitada. Por ello, el W3C desarrolló una generalización del CSS creándose XSL, que, como su propio nombre indica, es una extensión de las hojas de estilo.

W3C desarrolló estos tres lenguajes:

- **Xpath:** Lenguaje para poder acceder a partes de un XML.
- **XSLT (XSL Transformation):** Con este lenguaje, se podrá transformar el documento XML en otros documentos.
- **XSL-FO (XSL Formatting Objects):** Permite la transformación en formatos como pdf, o formatos de impresora.



Fig. 2. Introducción.

En este tema, nos centraremos en el aprendizaje de **XPath**.

Con este lenguaje, accederemos a los **nodos** que componen un documento XML, extraer sus datos, visualizar ciertos elementos e incluso hacer cálculos con nodos que contengan información numérica.

Al igual que ocurre con las versiones de HTML o XML, este lenguaje también tiene varias versiones, siendo la última la 3.1, que puedes consultar en este [enlace](#).

Un concepto a considerar antes de ahondar sobre el lenguaje, es el del **árbol del documento XML**.

Es decir, un documento XML consta de nodos y ramas que están interconectadas.

Por lo tanto, se puede considerar que el documento XML es un árbol dirigido donde los nodos están conectados por un único camino y sentido. Además, todos los nodos tienen un único padre, salvo el nodo raíz, que no tiene padre.

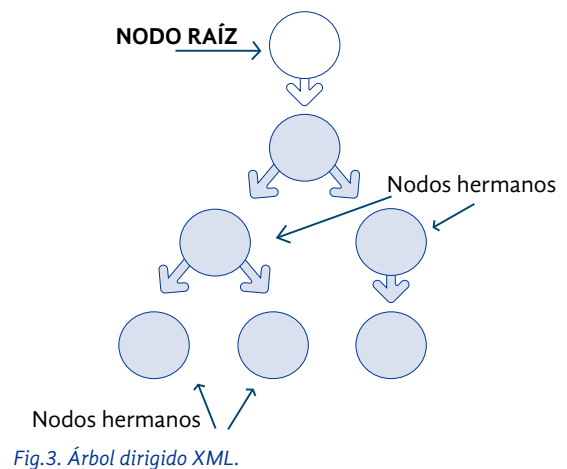


Fig.3. Árbol dirigido XML.

/ 3. El árbol: Nodos ascendientes y descendientes

Se consideran nodos **descendientes** a todos aquellos que descienden de él. Los nodos **ascendientes** son todos aquellos que son antecesores del nodo, o sea, el padre, el padre del padre, etc.

Es importante diferenciar claramente estos **conceptos** porque a la hora de recorrer el árbol y extraer información tendremos que hacer referencia a nodos ascendientes o hermanos de los ascendientes, etc.

A lo largo del tema, vamos a trabajar con el fichero XML que podemos ver en el **anexo** del tema.

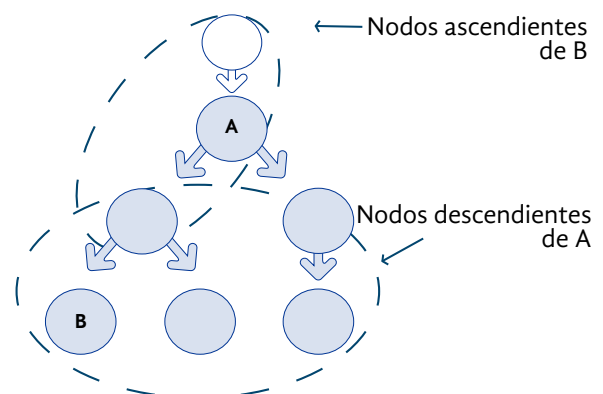


Fig.4. Nodos ascendientes y descendientes.

- **Tipos de nodos:** Cuando se relaciona el documento XML con un árbol dirigido, se obtienen estos tipos de nodos:
 - » **Raíz:** Nodo raíz del fichero XML. No confundir con el elemento raíz del documento que está por debajo de él. Se identifica con “/”.
 - » **Elemento:** Cada uno de los elementos (etiquetas) del XML.
 - » **Atributo:** Atributo asociado a un nodo elemento.
 - » **Texto:** Contenido del elemento. No contienen hijos.

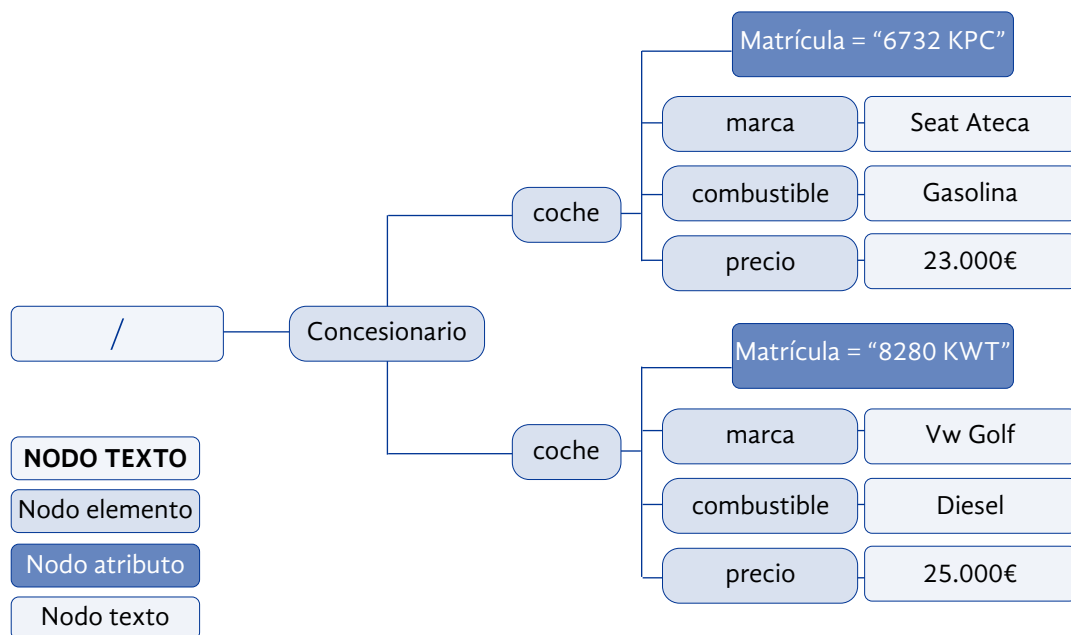


Fig. 5. Tipos de nodos de un documento XML con datos de un concesionario de coches.

/ 4. Sintaxis XPath. Eje

Las expresiones XPath son cadenas de texto que representan caminos y nodos del árbol XML. Tiene mucha similitud con las rutas de los archivos o carpetas que podemos tener en explorador de archivos de un sistema operativo. Estas expresiones se evalúan y pueden devolver un conjunto de nodos que coinciden con lo que se busca, un valor calculado, o ningún nodo si no hay coincidencias.

Las expresiones XPath funcionan en pasos de búsqueda y cada paso se divide en tres partes:

- **Eje:** Permite seleccionar elementos o atributos atendiendo a su nombre.
- **Predicado:** Permite restringir los nodos obtenidos con el eje aplicando condiciones.
- **Selección de nodos:** Permite indicar qué resultado final mostramos. Por ejemplo, el elemento completo, el contenido o los dos.



Audio 1. "XPath"

<https://bit.ly/2DhAFzo>



Comencemos el estudio de cada parte con el **eje**. Con este, vamos a obtener subconjuntos de nodos y corresponde a recorridos del árbol. En las siguientes tablas, se muestran los diferentes tipos de búsquedas y algunos ejemplos.

SINTAXIS	DESCRIPCIÓN
nombre_nodo	Selecciona todos los nodos con ese nombre.
/	Selección de nodos desde el nodo raíz.
//	Selecciona nodos desde el nodo actual sin importar donde se encuentren.
.	Selecciona el nodo actual.
..	Selecciona el padre del nodo actual.
@	Selecciona atributos.
	Permite indicar varios caminos.

Tabla 1. Selección de nodos en el eje.

XPATH	RESULTADO
/concesionario/coche /marca	<marca>Seat Ateca</marca> <marca>Volkswagen Golf</marca>
/marca	Nada-> marca no es hijo del nodo raíz.
/concesionario/marca	Nada-> marca no es hijo de concesionario.
/concesionario/coche /@matricula	matricula="6732KPC" matricula="8280KWT"
//marca	<marca>Seat Ateca</marca> <marca>Volkswagen Golf</marca>
//@matricula	matricula="6732KPC" matricula="8280KWT"
//marca //precio	<marca>Seat Ateca</marca> <precio>23.000€</precio> <marca>Volkswagen Golf</marca> <precio>25.000€</precio>

Tabla 2. Ejemplos de recorridos.

/ 5. Caso práctico 1: “Instituto de FP: Árbol”

Planteamiento: Un instituto se ha puesto en contacto con [WebMalagaDesign](#) porque tiene muchos datos administrativos que analizar y necesita tratarlos automáticamente.

Nudo: Lo primero que quiere hacer Alberto es un árbol del fichero XML para tener más claro la organización del instituto. Para ello, hay que analizar el fichero XML que podemos consultar en el anexo del tema. Se lo comenta a Gloria y le dice que por ahora solo haga el árbol de los dos primeros alumnos del ciclo de DAW, empezando como nodo raíz el módulo.

Desenlace:

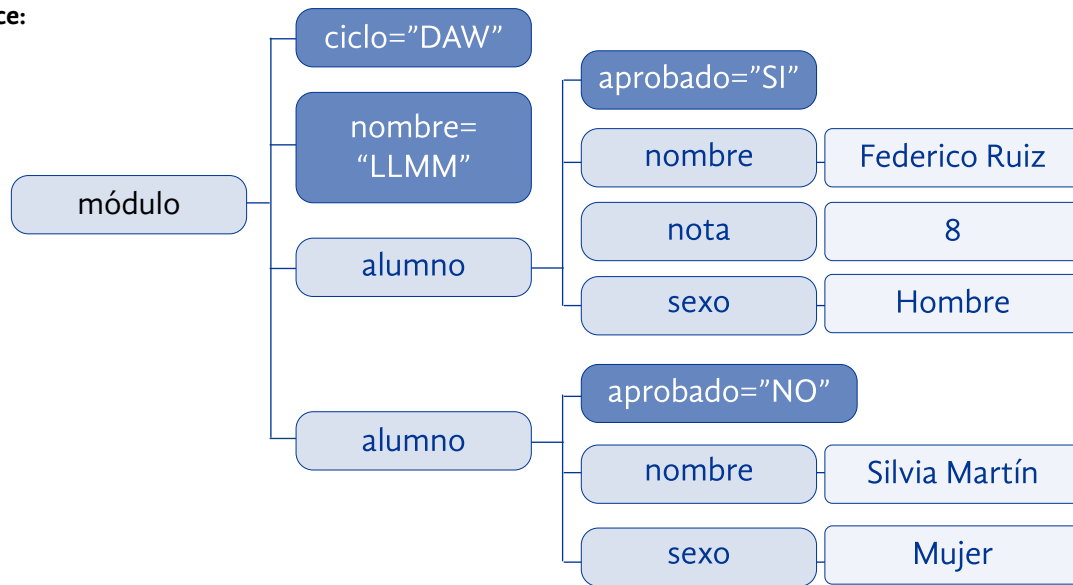


Fig. 6. Árbol del fichero XML del instituto.

/ 6. Sintaxis XPath. Predicado

Esta nueva parte de los pasos de búsqueda a seguir se tiene que escribir entre corchetes y nos va a permitir afinar la búsqueda aplicando condiciones. Se ha de escribir a continuación del eje. En la siguiente tabla, se listan las restricciones:

XPATH	RESULTADO
<code>//coche[2]@matricula</code>	<code>matrícula="8280KWT"</code>
<code>//coche [@matricula="6732KPC"]/precio</code>	<code><precio>23000</precio></code>
<code>//coche[precio=23000]</code>	<code><coche matricula="6732KPC"> <marca>Seat Ateca</marca> <combustible>Gasolina</combustible> <precio>23000</precio> </coche></code>
<code>//coche [combustible="Diesel" and precio>22000]/marca</code>	<code><marca>Volkswagen Golf</marca></code>
<code>//coche[precio>21000] [combustible="Gasolina"] /@matricula</code>	<code>matrícula="6732KPC"</code>
<code>//coche[precio*2>46000] /marca</code>	<code><marca>Volkswagen Golf</marca></code>
<code>//coche[precio*2>46000] /@matricula</code>	<code>matrícula="8280KWT"</code>
<code>count(//coche [combustible="Gasolina"])</code>	1
<code>sum(//precio)</code>	48000
<code>substring(//coche [@matricula="6732KPC"] /@matricula,4,1)</code>	K

Tabla 3. Restricciones en el predicado.

RESTRICCIÓN	DESCRIPCIÓN
número	Cuando se obtiene varios nodos, se muestra el indicado con el número.
@atributo	Selecciona los nodos que tiene el atributo.
función	Funciones que se pueden aplicar a los nodos obtenidos. Existen funciones de: Manipulación de datos: position (), last (), etc. Cadenas de caracteres: concat (), substring (), etc. Operaciones numéricas: sum (), count (), etc. Funciones booleanas: not (), etc. Funciones de fecha: current-date (), etc.
condición	Selecciona los nodos que cumplen la condición. La condición se puede hacer sobre atributos o valores de los elementos. En las condiciones, se pueden utilizar operadores lógicos (and, or, not), aritméticos (+, -, *, div, mod) y de comparación (=, !=, <, >, <=, >=).

Tabla 4. Ejemplos de predicados.

Existen multitud de funciones que se pueden utilizar. Para una referencia completa, puedes consultar el siguiente [enlace](#).



Vídeo 1. "Datos demográficos"

<https://bit.ly/3lwEAcM>



/ 7. Sintaxis XPath. Selección de nodos

La selección se indica después del eje y del predicado. Nos va a permitir seleccionar las partes del nodo que queremos mostrar.

SELECCIÓN	DESCRIPCIÓN
node ()	Selecciona todos los hijos (elementos o texto).
text ()	Selecciona únicamente el texto contenido y no muestra las etiquetas.
*	Selecciona todos los hijos (solo elementos).
@*	Selecciona todos los atributos del nodo.

Tabla 1. Tabla 5. Selección de nodos.

XPATH	RESULTADO
<code>//coche[@matricula="6732KPC"] /node()</code>	<code><marca>Seat Ateca</marca></code> <code><combustible>Gasolina</combustible></code> <code><precio>23000</precio></code>
<code>//coche/marca /text()</code>	Seat Ateca Volkswagen Golf
<code>//coche//*</code>	<code><marca>Seat Ateca</marca></code> <code><combustible>Gasolina</combustible></code> <code><precio>23000</precio></code> <code><marca>Volkswagen Golf</marca></code> <code><combustible>Diesel</combustible></code> <code><precio>25000</precio></code>
<code>//coche/@*</code>	<code>matricula="6732KPC"</code> <code>matricula="8280KWT"</code>

Tabla 1. Tabla 6. Ejemplos de selección de nodos.

La selección de nodos es de suma importancia porque, con ella, conseguimos llegar al elemento concreto que queramos mostrar de la consulta.

Habrà que tener especial cuidado con la selección en estructuras de datos complejas.

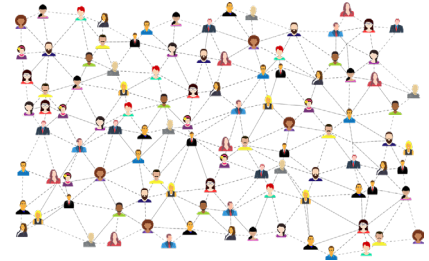


Fig. 7. Precaución en la selección de nodos.

/ 8. Sintaxis XPath. Expresiones anidadas

Las expresiones XPath se pueden anidar obteniéndose consultas más detalladas. En este caso, lo que hay que tener en cuenta es que el resultado de una expresión Xpath, puede formar parte del predicado de otra expresión. Como ejemplo de expresión anidada, se podría plantear que queremos obtener los coches cuyo combustible sea diferente al combustible del último coche que ha llegado al concesionario. Se podría resolver por partes.

- Todos los coches.

```
//coche
```

- Combustible del último coche.

```
//coche[last()]/combustible
```

- Se comparan los coches para que el combustible sea diferente al del último coche.

```
//coche[combustible!=//coche[last()]/combustible]
```

El resultado que se obtiene es:

```
<coche matricula="6732KPC">
  <marca>Seat Ateca</marca>
  <combustible>Gasolina</combustible>
  <precio>23000</precio>
</coche>
```

Código 1. Expresión XPath anidada.



Fig. 8. Las muñecas Matryoshka representan a la perfección el concepto de anidamiento.

Que es el único coche que es de Gasolina y, por lo tanto, con combustible diferente al del último coche que es Diesel.



Vídeo 2. "Consultas XPath – Datos demográficos– Parte 2"

<https://bit.ly/2GcC4s6>



/ 9. Caso práctico 2: “Instituto de FP: Consultas”

Planteamiento: Continuando con el caso del instituto que se ha puesto en contacto con [WebMalagaDesign](#) porque tiene muchos datos administrativos que analizar y necesita tratarlo automáticamente, se nos plantea otra cuestión.

Nudo: Gloria habla con el administrativo del instituto y este le comenta que la información le llega desde la consejería en un formato XML, que ellos no saben tratar y necesitarían obtener una serie de datos. Gloria le indica que no se preocupe y que les pase las consultas que necesitan.

Desenlace: En la siguiente tabla, se listan las consultas necesarias, la expresión XPath asociada y el resultado obtenido.

El fichero XML lo puedes encontrar en el **anexo** del tema.

CONSULTA	EXPRESIÓN XPATH	RESULTADO
Nombre del instituto	/instituto/@nombre	nombre="Medac Formación"
¿Qué ciclos se imparten de la familia de informática?	//especialidad[@nombre="Informática"] /ciclo/text()	Desarrollo de aplicaciones web. Administración de sistemas informáticos y redes. Sistemas microinformáticos y redes.
Número de ciclos superiores	count(//ciclo [@grado="Superior"])	3
Último ciclo de la familia de Informática sin la etiqueta ciclo.	//especialidad[@nombre="Informática"] /*[last()]/text()	Sistemas microinformáticos y redes.
Número de aprobados en LLMM	count(//modulo[@nombre="LLMM"]/alumno[@aprobado="SI"])	2
Nota de “Silvia Martín” en PRO, con un texto introductorio.	concat("La nota de Silvia es ", //modulo[@nombre="PRO"]/ alumno[nombre="Silvia Martín"]/ nota)	La nota de Silvia es 6.
Lista de clase de LLMM	//modulo[@nombre="LLMM"]/ alumno/nombre /text()	Federico Ruiz. Silvia Martín. Manolo Chicón.

Tabla 7. Consultas resolución caso práctico 2.

/ 10. Herramientas de validación de expresiones

Las expresiones XPath se pueden utilizar en lenguajes de programación como JavaScript, Java, PHP, C, C++ y muchos más. Para ello, simplemente se usará una sintaxis propia del lenguaje que permita incluir dicha expresión Xpath y nada más. Estas expresiones también se utilizan en las transformaciones XSL (XSLT), y que veremos en detalle en el próximo tema.

Antes de incluir estas expresiones en el lenguaje de programación con el que se esté trabajando, es conveniente poder analizar y comprobar si las expresiones XPath son correctas y que devuelven lo que se necesita.

Para ello, existen numerosas herramientas que nos ayudan en esta labor:

- **XML Copy Editor:** Dentro de las herramientas XML de este software, hay una opción que nos permite evaluar expresiones XPath. Simplemente se carga el fichero XML y luego se indica en una ventana la expresión XPath a validar, mostrándose el resultado.

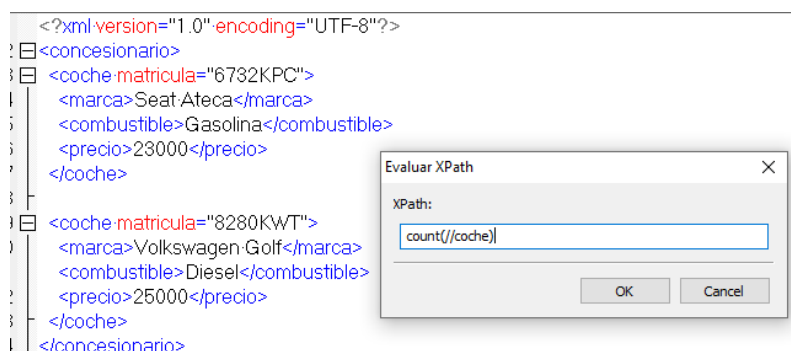


Fig. 9. Expresiones XPath en XML Copy Editor.

- **Notepad++:** Este software permite instalar un plugin llamado XML Tools que, entre otras muchas funcionalidades, puede evaluar expresiones XPath y también permite obtener el XPath del nodo en el que nos encontremos ubicados en el XML.

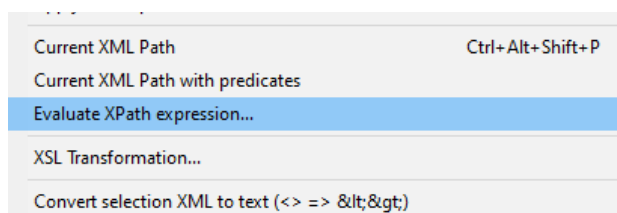


Fig. 10. XML Tools en Notepad++.

- codebeautify.org: Herramienta online que permite validar la expresión XPath y mostrar el resultado de manera inmediata.



Fig. 11. Expresión XPath a evaluar.

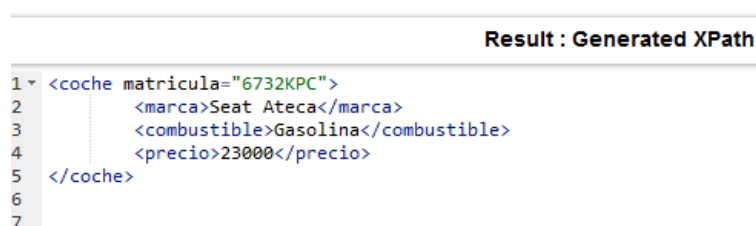


Fig. 12. Resultado obtenido en codebeautify.org

/ 11. Resumen y resolución del caso práctico de la unidad

En este tema se ha introducido el concepto de **transformación** de un documento XML y los **lenguajes** a usar para ello. Es muy importante conocer cómo está estructurado un XML: como un **árbol** dirigido. XPath va a permitir **recorrer el XML** y seleccionar y adaptar los contenidos que se quieren obtener. Por último, se han comentado algunas **herramientas** para trabajar con XPath.

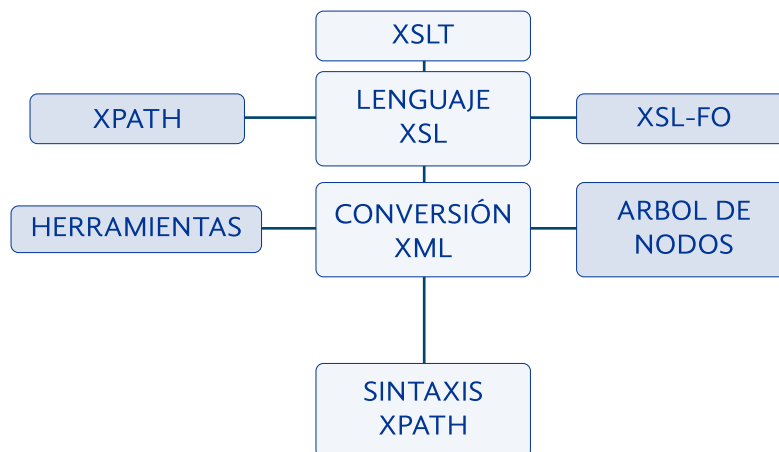


Fig. 13. Esquema del tema.

Resolución del caso práctico de la unidad

El fichero XML con las previsiones meteorológicas lo puedes consultar en el **anexo** del tema.

CONSULTA	EXPRESIÓN XPATH	RESULTADO
¿Qué previsión se tiene en Málaga para el día 11/07/2020?	<code>//poblacion[@nombre="Málaga"] [fecha="11/07/2020"]/ prevision/text()</code>	viento
¿Cuántos días de previsión se tienen de Málaga?	<code>count(//poblacion [@nombre="Málaga"])</code>	2
Mostrar la temperatura máxima y mínima de Bilbao el día 9/07/2020	<code>//poblacion[@nombre="Bilbao"] [fecha="09/07/2020"]/maxima //poblacion[@nombre="Bilbao"] [fecha="09/07/2020"]/minima</code>	<pre><maxima> 18 </maxima> <minima> 14 </minima></pre>
¿Cuántas poblaciones tendrán nubes cualquier día?	<code>count(//poblacion [prevision="nubes"])</code>	2
¿Qué población tendrá 35 grados de máxima cualquier día?	<code>//poblacion [maxima=35] /@nombre</code>	nombre="Madrid"

Tabla 8. Consultas sobre previsiones meteorológicas.

/ 12. Bibliografía

12.1. Webgrafía

[XML Copy Editor](#): Editor XML que permite también validar expresiones XPath.

[Notepad++](#): Editor HTML/CSS/XML que permite también validar expresiones XPath.

[codebeautify.org](#): Testeador online de expresiones XPath.

[Referencia XPath en W3C](#).