



Almacenamiento de información

Lenguaje de Marcas y
Sistemas de Gestión de
la Información



Índice



6.1. Sistemas de almacenamiento de información

- 6.1.1. Ficheros
- 6.1.2. Bases de datos XML habilitadas o enabled
- 6.1.3. Bases de datos XML nativas

6.2. Inserción y extracción de información en XML

6.3. Búsqueda de información en documentos XML: lenguajes de consulta y manipulación

- 6.3.1. XPath
- 6.3.2. XQuery

6.4. Almacenamiento XML nativo

- 6.4.1. Instalación y ejecución de una base de datos XML nativa: eXist-db
- 6.4.2. BaseX

6.5. Herramientas de tratamiento y almacenamiento de información en formato XML



Introducción

En este tema veremos las bases de datos, herramientas que nos permiten almacenar datos en formato XML, con el fin de que puedan ser consultados posteriormente con la mayor facilidad posible.

Con el fin de estudiar un elemento tan importante comenzaremos viendo los diferentes tipos de bases de datos existentes, distinguiéndolos por la estructura en la que trabajan. Una vez seleccionado el tipo de base de datos deseada, en

nuestro caso las bases de nativas, pasaremos a estudiar los lenguajes usados para trabajar con ellas, como son XPath o XQuery.

Por último, veremos cómo trabajar con bases de datos, comenzando desde su instalación y terminando por la enumeración de las herramientas que podemos usar para trabajar con ellas.

Al finalizar esta unidad

- + Conoceremos en mayor profundidad el lenguaje XML y las posibilidades que este nos puede ofrecer.
- + Descubriremos cómo instalar una base de datos No-SQL y las características que estas nos ofrecen.
- + Distinguiremos dos lenguajes de consulta y extracción de información como son XPath y XQuery y estudiaremos su uso.
- + Sabremos cómo instalar una base de datos eXist-db, y se indicarán las herramientas con las que podrás trabajar sobre él.



6.1.

Sistemas de almacenamiento de información

La información XML posee múltiples maneras de almacenamiento, como son:

- > En **ficheros**.
- > En **XML enabled**, bases de datos no nativas.
- > En **bases de datos nativas XML**.

6.1.1. Ficheros

Aunque es uno de los sistemas usados para el almacenamiento su uso no se recomienda, ya que no puede mantener los estándares de los demás sistemas de almacenamiento en materias como la concurrencia.

6.1.2. Bases de datos XML habilitadas o enabled

Las bases de datos XML enabled, también conocidas como bases de datos no nativas, son bases de datos relacionales, es decir bases que transforman en esquemas relacionales los archivos XML para, después, volcar esos datos en el propio esquema.

Este tipo de base de datos no requiere de middleware para trabajar con XML, hasta el punto de poder aceptar XML como formato de entrada y generar este mismo como salida.

Sus **desventajas** son las siguientes:

- > Ante futuros cambios, es imposible restaurar el documento original.
- > Problemas con la estructura, ya que este tipo de base es relacional mientras que el XML es jerárquico.

6.1.3. Bases de datos XML nativas

Se basa en XML y usa el documento XML como unidad de almacenamiento.

A diferencia de las bases XML enabled las bases nativas no suelen crear problemas, aunque los documentos XML sean extensas, y, además, como no llega a modificar los documentos XML, estos pueden ser recuperados en su estado original sin problemas.

Otra característica que le permite trabajar con una mayor calidad es el hecho de que este tipo de base de datos, al igual que los documentos XML, trabajan de manera jerárquica, lo que evita tener que realizar transformaciones que pueden desencadenar ciertos fallos además de gastar tiempo.

Los entornos nativos usan lenguajes como XQuery 1.0 y XPath 2.0.

Algunas de las bases de datos más utilizadas son:

- > **Oracle Berkeley DB XML**: De código abierto.
- > **BaseX**: Con licencia BSD.
- > **Sedna**: Con licencia LGPL.
- > **eXist**: Con licencia Apache.





6.2.

Inserción y extracción de información en XML

En la hoja de cálculo, en el suite de Google Docs, podemos encontrar una herramienta muy útil para cualquier programador es ImportXML. La ventaja de esta herramienta sobre otras es que trabaja con formatos estructurados lo cual permite importar los diversos formatos que usan este sistema, como son, entre otros, XML, HTML y CSV.

La sintaxis de ImportXML se basa en la siguiente estructura:

```
IMPORTXML("url"; "consulta_xpath")
```

La url, entre comillas, hace referencia a la página que desea examinarse, mientras que consulta_xpath, también entre comillas, hace referencia a lo que desea consultarse en la página ya mencionada en lenguaje XPath.

Como, por ejemplo:

```
IMPORTXML("https://www.w3schools.com/xml/default.asp"; "//a/@href")
```

```
IMPORTXML(A2; B2)
```

Con estas instrucciones es posible extraer información o estructuras de una página o archivo. Es un recurso imprescindible cuando se manejan grandes cantidades de datos con un estilo similar, como pueden ser las suscripciones o membresías, de las cuales podremos extraer información como números de teléfono, sin pasar por todo el listado de miembros.





6.3.

Búsqueda de información en documentos XML: lenguajes de consulta y manipulación

Los lenguajes de consulta más utilizados son el XPath y el XQuery, aunque existen otros más sofisticados y con mayores prestaciones como XPointer

6.3.1. XPath

XPath, un lenguaje path language, con una sintaxis path like que permite buscar y seleccionar nodos en archivos con lenguajes XML. XPath posee más de 200 comandos predefinidos que podemos manejar para desarrollar la función deseada. Además de ser un elemento importante con XSLT XPath es el recomendado por W3C. Podéis encontrar toda la información necesaria en: https://www.w3schools.com/xml/xpath_intro.asp

XPath puede trabajar con los lenguajes JavaScript, Java, XML Schema, PHP, Python, C y C ++, entre otros.

XPath analiza el documento XML con el fin de crear árboles de nodos de siete tipos:

- > Nodo raíz.
- > Elemento.
- > Texto.
- > Atributo.
- > Namespace (espacio de nombres).
- > Instrucción procesable.
- > Comentario.

Así, con archivo XML como el siguiente, podemos realizar diversas acciones y ver sus diferentes partes. Usaremos este XML como ejemplo para el resto del tema.

```
<?xml version="1.0" encoding="UTF-8"?>
<Almacén>
  <películas>
    <título lang="es">Harry Potter</título>
    <director>J K. Rowling</director>
    <año>2005</año>
  </películas>
  <películas>
    ...
  </películas>
  ...
</Almacén>
```



XPath permite trabajar un documento con estructura arbórea, como se ve en la siguiente imagen:

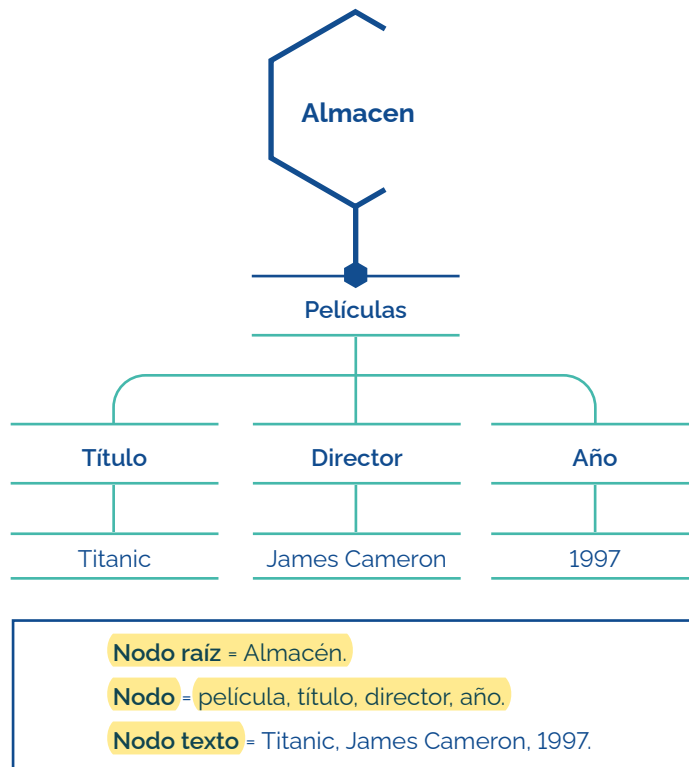


Imagen 1. Forma arbórea.

Los nodos en XPath

Dentro de la estructura los nodos se relacionan entre sí:

- > **Nodo.** Los valores intermedios.
- > **Nodo raíz.** El nodo principal es aquel que inicia el archivo.
- > **Valor atómico.** Elementos sin padres ni hijos.
- > **Ítem o elemento.** Puede ser tanto un nodo como un valor atómico.

Las relaciones entre los nodos son las siguientes:

- > **Padre (parent).** Es el nodo del que desciende otro, "películas" es padre de "título", "director" y "año".
- > **Hijo (children).** Es el nodo que desciende de otro, "título" es hijo de "película".
- > **Hermanos (siblings).** Nodos que comparten un mismo padre.
- > **Antecesoros (ancestors).** Es el nodo padre de otro nodo padre, el abuelo; así el elemento "Almacén" es el antecesor de "título".
- > **Descendientes (descendant).** Es el nodo hijo de otro nodo hijo, "título" es descendiente de "Almacén".



Cómo seleccionar los nodos de un documento XML

El lenguaje XPath usa expresiones de ruta para navegar por el documento:

Expresiones de ruta	Descripción
Nombre_deL_nodo	Selecciona los nodos con el nombre especificado
/	Selecciona el nodo raíz
//	Selecciona los nodos que coinciden con el nodo actual
.	Selecciona el nodo actual
..	Selecciona el nodo padre del actual
@	Selecciona atributos

Imagen 2. Expresiones de ruta.

Ejemplos:

Expresiones de ruta	Descripción
director	Selecciona los nodos "director"
/Almacén	Selecciona el nodo raíz
//título lang="es"	Selecciona los nodos «título lang="es"»
//director //año	Selecciona los nodos "director" y "año"
//@lang	Selecciona el atributo "lang"

Imagen 3. Expresiones de ruta ejemplos.

Así la siguiente ruta:

path = "/Almacén/películas/director"

Selecciona los nodos director hijos de películas, mostrándose solo estos de la siguiente manera:

director: James Cameron

director: ...2

director: ...3



Utilizando predicados para mejorar las búsquedas

Los predicados permiten extraer un nodo concreto de un conjunto, se escriben entre corchetes.

Expresiones	Significado
/Almacén/peliculas[1]/director	Se selecciona el director del nodo de películas 0, en este caso James Cameron. En ocasiones, dependiendo del navegador, los nodos se empiezan a contar desde el 0.
/Almacén/peliculas[last()]/director	El director de la última película.
/Almacén/peliculas[last()-1]/director	El director de la penúltima película.
//titulo[@lang="es"]	Solo los títulos de películas en español (atributo "es").
/Almacén/peliculas[año>1990]/director	Los directores de las películas del año 1991 en adelante.
/Almacén/peliculas[position()<3]/director	Los directores de las dos primeras películas.

Imagen 4. Predicados XPath.

Algunos de los operadores empleados por XPath son los siguientes:

Operador	Descripción	Ejemplo
	Computa dos nodos simultáneos	//director //año
+	Adición	1 + 1
-	Sustracción	3 - 1
*	Multiplicación	1 * 3
div	División	4 div 2
=	Igual	año=1997
!=	No igual	año!=1997
<	Menos que	año<1997
<=	Menos o igual que	año<=1997
>	Más que	año>1997
>=	Más o igual que	año>=1997
or	O	año=1997 o año=1996
and	Y	año>1990 y año<2000

Imagen 5. Operadores usados por XPath.



Combinar varias rutas

Con el fin de combinar varios días podemos usar el operador "|" que nos permitirá unir varias rutas, por ejemplo:

`path = "Almacén/películas/director | Almacén/películas/año"`

Mostrando así todos los directores y después todos los años.

Comodines

Para seleccionar nodos desconocidos XPath cuenta con expresiones específicas llamadas comodines:

Comodín	Descripción
*	Selecciona cualquier nodo del elemento
@*	Selecciona cualquier atributo del nodo
node()	Selecciona cualquier nodo de cualquier clase
Ejemplos	
<code>/películas/*</code>	Selecciona todos los hijos de películas
<code>//*</code>	Todos los elementos del documento
<code>//titulo[@*]</code>	Selecciona el elemento título sin importar el atributo

Imagen 6. Comodines y sus ejemplos.

6.3.2. XQuery

XQuery es un lenguaje de consulta y extracción de información y atributos de archivos XML. Es un superconjunto de XPath, ya que usa las expresiones de XPath, con una sintaxis FLWOR. Posee una semántica semejante a SQL. Al igual que XPath, XQuery es recomendado por W3C.

Entre otras cosas XQuery permite:

- > Extraer información de XML.
- > Generar informes con datos resumidos.
- > Transformar datos XML a XHTML.
- > Buscar información específica en documentos web.

La sintaxis FLWOR es un acrónimo de For, Let, Where, Order by, Return.

- > **For:** Selecciona una secuencia de nodos.
- > **Let:** Enlaza una secuencia a una variable.
- > **Where:** Filtra los nodos.
- > **Order by:** Ordena los nodos.
- > **Return:** Devolver, lo que nos dará como resultado, se evalúa en todos los nodulos.



```
<html>
<body>
  <h1>
    Almacenaje de filmpedia</h1>
  <ul>
    <script>
      { for $x in doc("Inventario21.xml")/Almacén/películas
        where $x/año>1990
        order by $x/título Lang="es"
        return <li>{data($x/título)}</li>
      }
    </script>
  </ul>
</body>
</html>
```

Nos dará como resultado el siguiente:

```
<html>
<body>
  <h1>
    Almacenaje de filmpedia:</h1>
  <ul>
    <li>El silencio de los corderos</li>
    <li>El cabo del miedo</li>
    <li>Tacones lejanos</li>
    <li>La bella y la bestia</li>
  </ul>
</body>
</html>
```

Donde la lista que veríamos sería:

Almacenaje de filmpedia:

El silencio de los corderos

El cabo del miedo

Tacones lejanos

La bella y la bestia



6.4.

Almacenamiento XML nativo

A continuación, veremos cómo se instala una base de datos XML nativa.

6.4.1. Instalación y ejecución de una base de datos XML nativa: eXist-db

Dentro de las bases de datos, que no usan SQL, una de las más difundidas es eXist-db.

Es de fácil instalación, se requiere un sistema operativo Windows, MAC OS, o Linux y poseer Java 1.7 o posterior.

Podemos descargar la última versión, en una extensión .jar, disponible en la sección de descargas de exist-db.org. Debemos introducir el archivo en la carpeta donde se instalarán. Abriendo la consola de comandos escribiremos la siguiente instrucción:

```
java -jar exist-db-setup-xxx.jar
```

Donde xxx será el número de versión y corrección completo. Tras la apertura de interfaz gráfica, elegimos el directorio de dirección. Nunca debemos instalarlo en "Program Files" o "Archivos de Programa" en ambientes Windows ya que generará conflictos con los permisos del sistema operativo.

Elegiremos la carpeta de destino de almacenamiento de datos, es posible elegir la propia carpeta de instalación. Tras esto se nos pedirá que introduzcamos una contraseña para el administrador, es necesario que dicha contraseña sea segura y recordable. En la misma pantalla se nos pedirá la configuración de memoria del servicio. Se recomiendan estas:

Por último, se elegirán los paquetes para instalar, sino se conoce la materia es recomendable dejar las elecciones por defecto. La instalación comenzará tras esto.

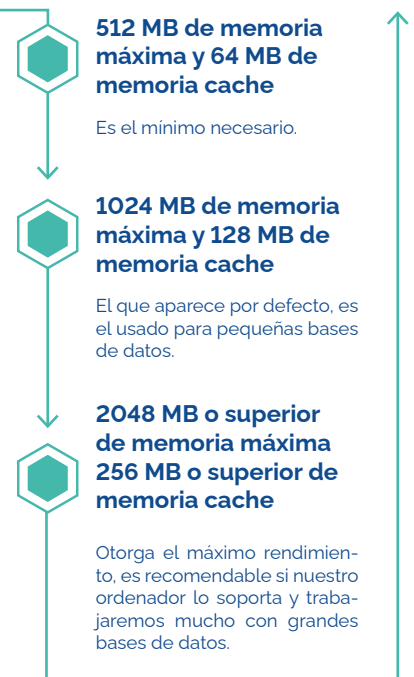
Para iniciar el servicio escribiremos en la consola de comandos el siguiente código:

```
java -jar start.jar
```

Los puertos de interfaz web que eXist-db usa son el 8080 para todo HTTP y el 8443 para HTTPS si se requiriese una interfaz SSL para nuestras aplicaciones.

6.4.2. BaseX

BaseX es una base de datos XML bastante fiable muy utilizada para el desarrollo de aplicaciones web por el hecho de tratarse de un código abierto. Podrás encontrarla en basex.org.





6.5.

Herramientas de tratamiento y almacenamiento de información en formato XML

Existen una multitud de herramientas que nos permiten trabajar con XML y HTML que ya hemos visto tanto en este tema como en temas anteriores es el caso de las herramientas de edición vistas en temas anteriores, como son Notepad, Gedit o los editores XML, y las herramientas de almacenamiento, las bases de datos, que ya hemos visto en este tema. Es destacable entre las herramientas que podemos usar eXide, ya que es la usada por eXist-db.

Es imprescindible encontrar las herramientas idóneas para nuestras necesidades, ya que no todas ellas poseen las mismas características o no son capaces de trabajar entre si sin conflictos.





 www.universae.com

