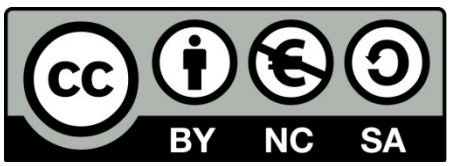
Curso básico de tecnologías XML

1. XML: conceptos y visualización de contenidos

INAP
INSTITUTO NACIONAL DE ADMINISTRACIÓN PÚBLICA

Contenido



1. Introducción.

XML nace como la necesidad de encontrar un lenguaje que sea capaz de estructurar la información y de ser compatible para las distintas plataformas existentes. No se puede considerar XML como un lenguaje de programación. Se trata de un metalenguaje con una gran potencia y versatilidad ya que nos permite crear nuestras propias etiquetas y visualizarlas en un navegador Web.

2. Objetivos.

En esta primera unidad se centrará en los siguientes puntos:

- Se realizará una introducción a XML comprendiendo el formato de los documentos XML.
- Se aprenderá a visualizar la información contenida en un navegador como Internet Explorer.
- Obtendremos un primer contacto con las posibilidades de visualización de documentos XML en navegadores (islas de información, enlace de elementos XML a elementos HTML, CSS, etc.).
- Comprenderemos el proceso por el que pasa un documento XML, desde que es escrito hasta que se visualiza en un navegador.

3. Introducción a XML.

Más que como un lenguaje en el término clásico de la palabra, **XML** se concibe como una herramienta′ de almacenamiento de datos que puede contener cualquier tipo de información. El primer anteproyecto de creación de XML es del año 1996. En este primer borrador, se concibe XML con una versión simplificada del lenguaje **SGML**, que permite al SGML genérico ser servido y procesado en la Web de la misma forma que HTML. SGML son las siglas de "*Standard Generalized Markup Language*" o lenguaje de marcación generalizado. Éste consiste en sistema para la organización y etiquetado de documentos. En dicho borrador, se considera un documento XML como un objeto de datos que se almacena en un computador y que está formado por diversos elementos y etiquetas.

Además, se explica la necesidad de usar un procesador para los documentos XML, que estará situado en una aplicación externa para la interpretación correcta del código. Posteriormente, se han recogido en diversos documentos las diferencias y mejoras que propone XML frente a HTML:

- XML permite a los proveedores de la información **crear nuevas etiquetas (elementos) y atributos**.
- XML permite a los documentos tener **niveles anidados de complejidad**.
- Con XML podemos **comprobar la validez de la estructura en tiempo real**.
- En términos de eficiencia, cuando los navegadores trabajan simultáneamente con varias fuentes de datos o de información, **XML es mucho más solvente**.

En el año 1997, **W3C** (*World Wide Consortium*) presentó XML 1.0 como recomendación. Actualmente, se ha ampliado la gama de tecnologías derivadas de XML. A continuación, enumeraremos las principales:

- **XHTML 1.0**: Recomendación de la organización W3C de Enero de 2000. Es la versión XML de HTML. Está pensado para sustituir a HTML como lenguaje estándar de páginas Web.
- **Lenguaje Extensible de Hojas de Estilo (XSL) 1.0**: Más que un lenguaje, es una familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera, debe ser tratada o transformada para su presentación en un medio específico.
- **Transformaciones de XSL (XSLT) 1.0**: Se trata de un estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML. Para ello, se utilizan diversas reglas o plantillas.
- **XLink**: Se trata del Lenguaje de vínculos XML. Es una recomendación del W3C que nos permite crear elementos de XML que describen relaciones entre documentos, imágenes y otros archivos de Internet.
- **XPath**: Se trata de un lenguaje que nos permite seleccionar un subconjunto de un documento XML. Permite buscar y seleccionar contenido teniendo en cuenta la estructura jerárquica del documento.
- **XML Schemas**: Es una recomendación del W3C que contempla la definición de un tipo de documentos o datos que sirven para especificar y ofrecer el contenido de cualquier documento. Su estilo es muy similar a los DTD (Document Type Definition).
- **XForms 1.0**: Se utiliza para la creación de interfaces de usuario y formularios Web. Su uso se está extendiendo notablemente con el auge de las aplicaciones Web. Hoy día es raro ver que algún navegador soporte XForms de forma nativa, ya que la recomendación es de Marzo de 2006, pero ya existen numerosos plug-ins para incluir XForms en los navegadores más usados (Internet Explorer, Mozilla Firefox).

En la siguiente imagen podemos ver el conjunto de **tecnologías** que forman XML:

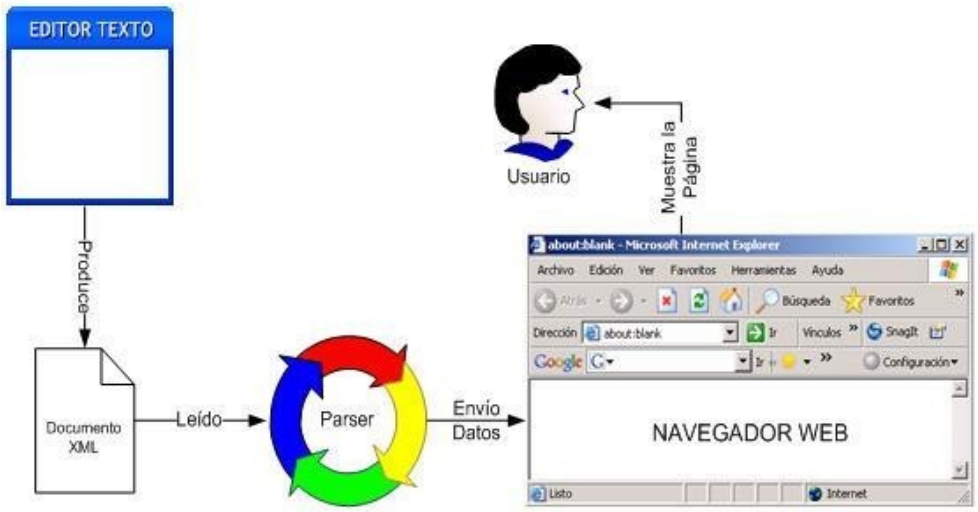
Las características más destacadas de XML son las siguientes:

- XML puede **almacenar y organizar cualquier tipo de información** y de la forma que más se adapte a nuestras necesidades.
- Debido a que se trata de un **estándar abierto**, no está ligado a una plataforma concreta o a un software determinado.
- XML **no es un solo producto, sino una familia de tecnologías**. Tal y como indican sus siglas, es extensible (*eXtensible*) y como metalenguaje, XML es capaz de crear sus propios lenguajes de marcas. Actualmente cuenta con una especificación Xlink, que describe un modo estándar de añadir hipervínculos a un documento XML. Además, el lenguaje de hojas de estilo (CSS) se puede utilizar con XML al igual que se hace con HMTL. El *Modelo de Objetos de Documento (DOM)* es un conjunto estándar de funciones para manipular documentos XML mediante un lenguaje de programación. *XML Namespaces*, es una especificación que describe cómo puede asociarse una URL a cada etiqueta de un documento XML, otorgándoles un significado adicional. Y finalmente, *XML-Schemas* es un modo estándar de definir los datos incluidos en un documento de forma más similar a la utilizada por los programadores de bases de datos, mediante los metadatos asociados.
- Posee una **estructura bien definida**: Cada documento XML consta de una raíz de la que descienden de forma anidada el resto de elementos. Con esta característica se consigue eliminar la desorganización que caracterizaba a HTML.
- XML está escrito en **formato de texto**. A pesar de ello, su objetivo está más cerca de organizar la información que de ser leído. Con esta característica, obtenemos numerosas ventajas acerca de la portabilidad e independencia de la plataforma. Su sintaxis es más estricta que HTML. No existe flexibilidad o

permisividad en la construcción de documentos para evitar así problemas más graves.

- XML ofrece **varias maneras de controlar la calidad de un documento**. Para ello, usa reglas de sintaxis, comparación con modelos de documentos, etc.
- Su **sintaxis clara y concisa** facilita la revisión, depuración y lectura del código.
- XML **no requiere licencias**. La selección de XML como soporte de aplicaciones, significa entrar en una comunidad muy amplia de herramientas y desarrolladores, facilitando así la universalidad del lenguaje.

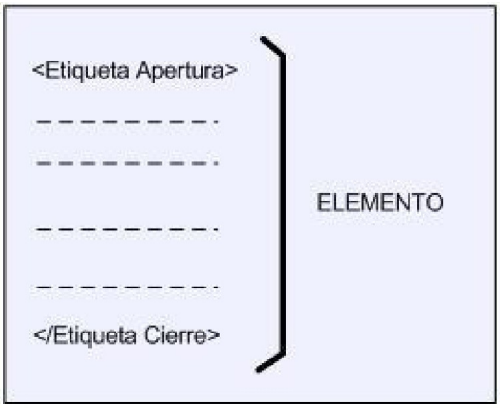
La imagen que vemos a continuación nos muestra el ciclo de vida clásico que sigue un documento XML.



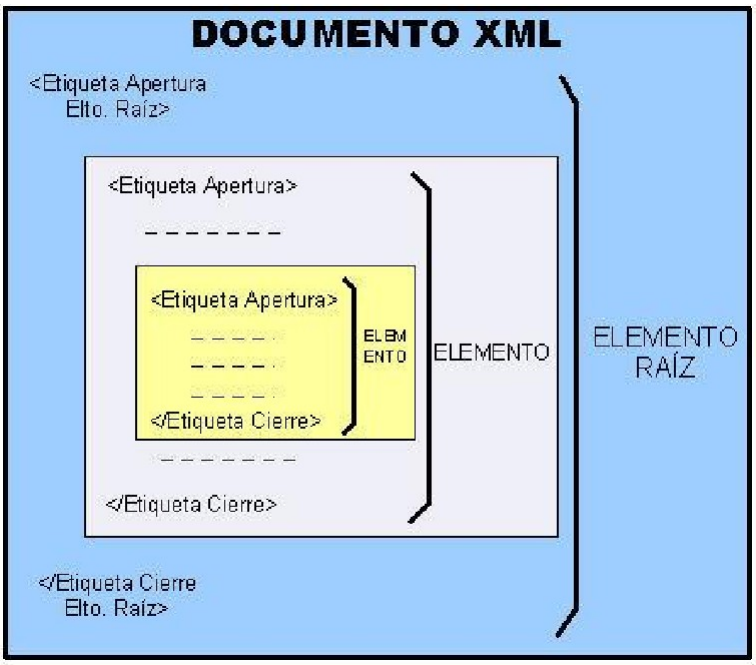
4. Examinando un documento XML.

Hasta ahora, cuando se escucha la palabra *documento*, es probable que pensemos en ésta como una secuencia de palabras organizadas en párrafos, secciones y capítulos con el objetivo de construir un artículo, libro, etc. Sin embargo, para XML el concepto de documentos debe ser entendido como algo más general. Se trata de la **unidad básica de información** que manejaremos.

Esta compuesto por *elementos* y *etiquetas*. Un elemento siempre debe contener dos etiquetas (una de apertura y una de cierre). Para entender la diferencia entre ambos conceptos, veamos la siguiente imagen:



Los elementos se van anidando como si fuesen cajas pequeñas que se introducen en cajas más grandes estructurando así el contenido del documento. En el nivel más alto tenemos el **elemento raíz** (*root element*). En la siguiente imagen se ilustra la organización de un documento en XML:



Veamos a continuación el ejemplo más sencillo en XML. Se trata del clásico ejemplo '*Hola Mundo*':

```
1  <?xml version="1.0" encoding="UTF-8" standalone= 'yes'?>
2  <!DOCTYPE Mensaje [
3  <!ELEMENT Mensaje (Contenido)>
4  <!ELEMENT Contenido (#PCDATA)>
5  ]>
6  <!-- este es un comentario -->
7  <Mensaje><Contenido>"¡Hola Mundo!"</Contenido></Mensaje>
```

Text | Grid | Schema/WSDL | Authentic | Browser

Comentamos a continuación cómo se descompone el ejemplo:

- La **primera línea**, constituye la definición general. Nos indica que lo que viene a continuación es un documento XML Además, observamos tres atributos (posteriormente veremos con detalle la definición de atributo):
  - **Version**: Indica el tipo de versión de la recomendación XML. Se establece a 1.0 e indica al intérprete de XML que debe de utilizar las normas establecidas en dicha recomendación (febrero 98).
  - **Encoding**: asignado a “UTF-8”, y que el estándar recomienda incluir siempre, aunque algunos navegadores (como Explorer 5) no lo exijan de forma explícita. Debemos tener en cuenta que XML debe soportar características internacionales, por tanto se dice que, tras su interpretación, todo documento XML devuelve Unicode. El valor por defecto es “UTF-8”, pero XML soporta los siguientes esquemas de codificación:
    - UTF-8
    - UTF-16
    - ISO-10646-UCS-2
    - ISO-10646-UCS-4
    - ISO-8859-1 a -9
    - ISO-2022-JP
    - Shift_JIS
    - EUC-JP
  - **Standalone**: Indica si el documento va acompañado de un DTD ("*no*"), o no lo necesita ("*yes*"). Su inclusión no es obligada, ya que en principio se indica si es necesario su uso o no, posteriormente en el DTD.
- La **segunda línea** es un DTD (Document Type Definition) muy simple. Más adelante se profundizará en este concepto, pero por ahora comentaremos que se utilizan para que un documento XML sea interpretado correctamente. Su definición se basa en una normativa rígida definida por XML. Esta DTD consta de la declaración del tipo de documento mediante !DOCTYPE seguido del nombre genérico que va a recibir el objeto que se defina a continuación (Mensaje), e indica que sólo va a contener un elemento (!ELEMENT) que también se denominará Mensaje y que está compuesto de otro elemento Contenido formado por texto (#PCDATA).
- La **tercera línea** nos muestra como se incluyen comentarios en el código XML.
- Por último, la **cuarta línea** contiene el elemento en sí, que contiene la información del documento. Para ello, es necesario incluir dos etiquetas de apertura y cierre con el nombre definido en la línea 2 (contenido). Aquí se incluye la información propiamente dicha. Si visualizamos el documento en un navegador Web obtendremos una salida como la siguiente:

```
<?xml version="1.0"?>
- <Mensaje>
    <Contenido>"¡Hola Mundo!"</Contenido>
  </Mensaje>
```

5. Estructura de un documento XML. Documentos bien formados.

El conjunto de todos los elementos de un fichero XML se denomina **documento**. En XML existen una serie de características que debe reunir todo documento para considerarlo documento bien formado. Esta denominación la recibe cualquier fichero que esté construido siguiendo las normas del estándar. A la hora de examinar la estructura de un documento XML, podemos diferenciar una estructura a nivel lógico y otra estructura a nivel físico.

Desde el punto de vista físico podemos considerar un documento XML como **un conjunto de unidades llamadas elementos**. Estos elementos pueden relacionarse con otros elementos para incluirlos en el documento.

Todos los documentos deben empezar por un elemento que se considera el '**elemento raíz**'. A parte de esto, cada documento podrá contener comentarios, declaraciones, atributos, etc. Tradicionalmente, se establece una división de los tipos de documentos en dos grupos: bien formados y válidos.

5.1 Documentos bien formados.

Son aquellos ficheros que han cumplido las especificaciones del lenguaje respecto a las reglas sintácticas, sin estar sujetos a unos elementos fijados en un DTD.

Como se ha comentado anteriormente, los documentos XML deben tener una estructura jerárquica muy estricta que los documentos bien formados deben cumplir. Existen algunas reglas básicas que deben cumplir:

- Todo elemento debe tener una **etiqueta de apertura y otra de cierre**.
- **Estructura jerárquica de elementos**: Se debe respetar una estricta jerarquía en lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente "incluida" en otra. Además, cualquier elemento con contenido, debe estar correctamente "cerrado". Podemos ver un par de ejemplos para ilustrar un caso correcto y un caso incorrecto.

```
<li>HTML <b> acepta <i> esto como válido </b> </i>.

<li>En XML la <b> estructura <i> cumple </i>

una jerarquía </b>.</li>
```

- **Etiquetas vacías**: XML permite al igual que HTML, introducir elementos sin contenido, pero la etiqueta debe ser de la siguiente forma:

<elemento sin contenido/>

Veamos otro par de ejemplos:

```
<li> HTML <br> :casi todo permitido </li>
<li> XML: <br/> es más restrictivo.</li>
```

- **Un solo elemento raíz**: Los documentos XML sólo permiten un elemento raíz, del que deben colgar todos los demás elementos del documento. Es decir, la jerarquía de elemento de un documento XML bien formado sólo puede tener un elemento inicial.
- **Valores de atributos**: Un valor de atributo en XML siempre deben estar encerrado entre comillas simples (′) o dobles ("). De los dos ejemplos que veremos a continuación, el primero de ellos no sería un ejemplo válido.

```
<a HREF=http://www.fi.upm.es/>
<a HREF="http://www.fi.upm.es/">
```

- **Tipos de letras, espacios en blanco**: El XML es **case sensitive**. Es decir, diferencia las palabras escritas en mayúscula o minúscula. Es decir, no es lo mismo *Private* que *PRIVATE* que *PrivaTE*. XML denomina a un conjunto de caracteres como "espacios en blanco" que son: "espacios", tabuladores, retornos de carro y saltos de línea. La especificación XML 1.0 permite el uso de esos "espacios en blanco" para hacer más legible el código, y en general son ignorados por los procesadores XML.
- **Nombrado de etiquetas**: Como se ha mencionado ya, XML permite a un usuario crear sus propias etiquetas. A pesar de esto, existen algunas restricciones en los nombres de dichas etiquetas. No están permitidas las que empiezan por la cadena "xml", "xML", "XML" o cualquier otra variante. Cualquier letra se puede usar en cualquier parte del nombre. También se pueden incluir dígitos, guiones y caracteres de punto, pero no se puede empezar por ninguno de ellos. El resto de caracteres, como algunos símbolos, y espacios en blanco, no se pueden usar. El siguiente fragmento de código XML sería un documento bien formado:

```
<?xml version="1.0"?>
<BIENVENIDO>
Hola XML!
</BIENVENIDO>
```

El código anterior tiene como etiqueta definida '*BIENVENIDO*'. Uno se puede preguntar viendo esto, que significa dicha etiqueta. La respuesta es: 'Significa lo que el usuario quiera'. De ahí la potencia de XML como metalenguaje. Al margen de las etiquetas que tiene ya definidas el lenguaje, podremos crear nuevas etiquetas para adaptarlas a nuestras necesidades. Con estas nociones básicas, deberíamos entender que el anterior código produciría el mismo resultado que el siguiente:

```
<?xml version="1.0"?>
<WELCOME>
Hola XML!
</WELCOME>
```

A continuación mostraremos algunas etiquetas correctas:

```
<HOLA>
 <curso>
 <Nombre1>
 <Nombre.Apellidos>
 <Nombre_Apellidos>
 <_1Nombre>
```

Y algunas etiquetas incorrectas:

```
<1HOLA>
<Nombre Apellidos>
<.Nombre_Apellidos>
< Nombre Apellidos >
```

- **Marcado y datos: l**as construcciones con etiquetas, referencias de entidad y declaraciones se denominan "*marcas*". Éstas son las partes del documento que el procesador XML entiende. Las marcas en un documento XML, son aquellas porciones que empiezan con "<" y acaban con ">", o bien, en el caso de las referencias de entidad, empiezan por "&" y acaban con ";".

5.2. Documentos Válidos

Son los documentos que cumplen la condición de estar bien formados y además, siguen una estructura y una semántica determinada por una **DTD** (*Document Type Definition*) o un **esquema XML**. Sus elementos y sobre todo la estructura jerárquica, deben ajustarse a lo que la DTD o el esquema dicten.

En definitiva, un documento XML **no puede ser válido si no está bien formado y no se puede analizar correctamente**. Cuando el documento es procesado por un parser que busca que se cumplan estas dos características (Bien formado y válido), se generará un error.

6. Visualizando Datos XML con Internet Explorer.

Para leer e interpretar correctamente XML, el navegador debe incluir un parser XML, que se encarga de realizar un chequeo previo de la fuente XML. Si al realizar ese chequeo se determina que el documento cumple los requisititos de validez y de documento bien formado, entonces el parser reestructura los datos para que puedan ser interpretados por la propia aplicación (en este caso el navegador Web).

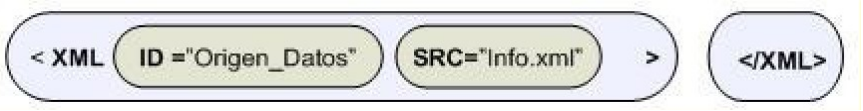
En el caso de incluir Definiciones de Tipo de Documento (DTD) o esquemas XML, se integran en este instante. Uno de los navegadores más extendidos es Internet Explorer, que desde su versión 4.0 soporta XML. Está provisto de dos parsers y soporta DHTML, CSS1, DOM1, SMIL, Microsoft XML 3.0 y .NET.

Para visualizar un documento XML en un explorador como Internet Explorer, simplemente hay que ejecutar el archivo *.xml en caso de que se tenga preestablecido la apertura de este tipo de ficheros con Internet Explorer. En caso contrario, habría que elegir la aplicación Internet Explorer para su apertura.

7. Islas de Información XML y Enlaces XML en Documentos HTML.

La técnica de islas de información o de datos, se incluye desde la implementación 4.0 y 5.0 de Internet Explorer, en la que el parser de HTML permite incluir una etiqueta ′< ′XML′>′ para integrar un documento escrito en XML. De esta forma, se le aplica un tratamiento separado del resto del código HTML. Lo más normal, es usar la etiqueta XML para realizar una referencia o enlace a documentos externos XML de modo que éstos sean cargados en memoria y procesados posteriormente.

La sintaxis sería la siguiente:



El procedimiento que sigue el parser es el siguiente:

- Se encuentra una **etiqueta XML** en un documento. A partir de aquí, asumimos que se trata de un conjunto estructurado de datos.
- Se crea en memoria un *recordset* con acceso de lectura a partir de la información contenida en el fichero *Info.xml*.
- El fichero *Info.xml* debe situarse en un directorio accesible por la página Web en cuestión.
- El *recordset*, con el nombre *Origen_Datos*, puede implementarse mediante algún lenguaje tipo JavaScript.

8. Conclusión.

En esta unidad didáctica se ha buscado que el alumno se familiarice con la versatilidad y la flexibilidad que nos ofrece XML. Hoy día es una de los lenguajes claves en la estructuración de la información y en la migración de datos entre plataformas distintas. Se ha profundizado en el concepto de lenguaje de etiquetas, así como en la estructura que sigue un **documento XML**, partiendo de un elemento raíz.

Los documentos XML deben apoyarse en una definición de tipo de documento que valida el código en base a una serie de declaraciones y convenciones. Además, se ha explicado el concepto dedocumento bien formado y de documento válido. El alumno sabe ahora que debe cumplir un documento XML para adaptarse a ambas definiciones.

9. Ejercicio Propuesto.

Crear un **documento XML** especificación 1.0 que no vaya acompañado de una DTD. Este documento debe almacenar información (a través del número de elementos y atributos que el alumno desee) bien organizada y con una estructura jerárquica sobre un dominio relacionada con los viajes (ocio y negocio). El documento debe tener una extensión aproximada de unas 50 líneas.