

UNIDAD DIDÁCTICA 4

DEFINICIÓN DE ESQUEMAS Y VOCABULARIOS EN LENGUAJES DE MARCAS

**MÓDULO PROFESIONAL:
LENGUAJES DE MARCAS Y SISTEMAS DE
GESTIÓN DE LA INFORMACIÓN**



CESUR
Tu Centro Oficial de FP

Índice

RESUMEN INTRODUCTORIO	2
INTRODUCCIÓN	2
CASO INTRODUCTORIO	3
1. DESCRIPCIÓN DE LA INFORMACIÓN TRANSMITIDA EN DOCUMENTOS XML. TECNOLOGÍAS PARA LA DEFINICIÓN DE DOCUMENTOS. ESTRUCTURA, SINTAXIS Y REGLAS.....	4
1.1 Descripción de la información transmitida en documentos XML	4
1.2 Tecnologías para la definición de documentos	5
1.3 Estructura, sintaxis y reglas	6
2. CREACIÓN DE DESCRIPCIONES DE DOCUMENTOS, UTILIZACIÓN DE MÉTODOS DE DESCRIPCIÓN DE DOCUMENTOS XML.....	10
3. ASOCIACIÓN DE DESCRIPCIONES CON DOCUMENTOS. VALIDACIÓN	14
4. HERRAMIENTAS DE CREACIÓN Y VALIDACIÓN.....	19
RESUMEN FINAL	25

RESUMEN INTRODUCTORIO

En esta unidad veremos las tecnologías que permiten la definición de documentos dentro de los lenguajes de marcas, conociendo los aspectos más importantes de cómo deberá ser su estructura y la sintaxis a utilizar.

Haremos un recorrido por las diferentes formas que puede recoger el documento XML a la hora de describir, por ejemplo, los elementos que en él se incluyen, así como la asociación a esas descripciones de un modo correcto, que permita la creación de documentos bien formados, permitiendo su validación.

Y, finalizaremos con un breve recorrido por los distintos tipos de herramientas que existen en el mercado para la creación y validación de un documento XML.

INTRODUCCIÓN

La importancia que tiene una correcta creación de un código fuente, radica en el hecho de que, a la hora de ser procesado, no dé ningún tipo de error.

Para los desarrolladores es tarea primordial conocer cómo debe implementarse el lenguaje que utilicen, como en este caso el lenguaje de marcas, para lo que deberán tener en cuenta las reglas y sintaxis de su creación.

Como parte del proceso, existe una manera de minimizar el esfuerzo realizado a la hora de escribir código fuente, y es, gracias a la utilización de distintas herramientas que permiten su edición (creación o modificación), imprescindibles en un mundo actual tan tecnológico.

CASO INTRODUCTORIO

Estás trabajando en el departamento de informática de una empresa que se dedica al desarrollo de software. En la actualidad te dedicas al desarrollo de aplicaciones web para un cliente que ha solicitado crear un sitio web corporativo relacionado con prendas de ropa deportivas.

Van a crearse una serie de páginas web que conformarán todo el portal de la empresa del cliente, para lo que utilizaréis documentos XML que, posteriormente, se procesarán en los navegadores web de los usuarios.

Para ello, tendrás que tener en cuenta las reglas y sintaxis para crear documentos XML bien formados con el fin de validarlos y que no existan errores a la hora de procesarlos en los distintos navegadores web.

Al finalizar el estudio de la unidad, conocerás cuáles son los esquemas que se utilizan en la creación de documentos XML, serás capaz de escribir código en un documento XML con una sintaxis correcta, realizarás el proceso de validación de documentos XML y utilizarás algún tipo de herramienta de edición de código para XML.

1. DESCRIPCIÓN DE LA INFORMACIÓN TRANSMITIDA EN DOCUMENTOS XML.

TECNOLOGÍAS PARA LA DEFINICIÓN DE DOCUMENTOS. ESTRUCTURA, SINTAXIS Y REGLAS

En el desarrollo del sitio web en el que estás trabajando, es requisito imprescindible conocer las tecnologías disponibles para una correcta definición de los documentos a elaborar, así como su estructura y sintaxis del tipo de código que se vaya a utilizar, en este caso XML.

Cuando hablamos de lenguajes de marcas, resulta inevitable que los primeros que nos vienen a la cabeza son HTML y XML. Y, en particular será XML el lenguaje de referencia por ser un metalenguaje que sirve de base en la construcción de otros lenguajes.

1.1 Descripción de la información transmitida en documentos XML

Ya hemos comentado que en XML no se especifican las etiquetas, pero sí se definen las reglas de construcción de documentos, en ellos deben definirse los elementos y atributos, que serán la descripción que se transmite mediante un documento XML.

Contará para ello con los siguientes componentes o nodos:

- Raíz, será el primer elemento que aparece en el documento designado con “/”.
- Elementos, son la unidad básica del documento, donde se almacena la información a transmitir, con sus correspondientes etiquetas.
- Atributos, asignan valores a los elementos, ampliando o describiendo la información sobre ellos o que queremos transmitir.
- Texto, es el grueso de la información dentro del documento en formato bien como contenido de un elemento, bien como valor de un atributo.
- Comentarios, con una funcionalidad idéntica a HTML.
- Espacio de nombres, permite diferenciar las etiquetas.
- Instrucciones de procesamiento, enviadas al procesador, comienzan con <? Y finalizan con ¿>.
- Entidades predefinidas, utilizadas para representar caracteres especiales de marcado.

- Secciones CDATA, son grupos de caracteres que requieren ser analizados por el procesador.
- DTD, recoge las reglas que se aplican al documento DTD.

1.2 Tecnologías para la definición de documentos

En la creación, distribución y manipulación de información a través de las tecnologías hacen necesario un soporte informático que le dé formato para que los documentos sean interpretados y permitan su lectura por un ordenador.

Los lenguajes de marcas son una forma de codificar un documento, incorporando una serie de etiquetas junto al texto o información, mediante una estructura y sintaxis determinadas.

Aportan como ventaja poder explicar el tipo de información que estamos almacenando, pudiendo darle el formato necesario y facilitar su procesamiento para que la información llegue al usuario final.



XML

Fuente: <https://www.danysoft.com/optimizacion-de-documentos-xml-grandes-parte-1/>

Vamos a centrarnos en la tecnología XML donde encontramos distintas tecnologías como son:

- XSL. Tiene por objetivo mostrar cómo debería estar estructurado el contenido para una correcta presentación en los distintos soportes de visualización.
- XPath. Lenguaje de rutas XML que permite el acceso a las distintas partes de los documentos XML.
- XLink. Lenguaje de enlace XML que permite insertar elementos en los documentos para crear enlaces entre recursos XML.
- XPointer. Lenguaje de direccionamiento XML que permite el acceso a la estructura interna de un documento XML: Elementos, atributos y contenido.

- XQL. Lenguaje de consulta XML que permite la extracción de datos de los documentos XML, disponiendo de opciones para la flexibilización de las consultas.

1.3 Estructura, sintaxis y reglas

Ante la aparente similitud entre un documento XML y un documento HTML, existe una diferencia fundamental, mientras HTML no indica lo que se está representando, en XML si se indican los datos describiendo el contenido de lo que se etiqueta.

Un ejemplo de la estructura de un documento XML sería:

```
<?xml version="1.0" encoding="UTF-7" ?>
<album>
  <autor>Alejandro Sanz</autor>
  <titulo>Más Aniversario</titulo>
  <formato>MP3</formato>
  <localizacion>Varios CD2 </localizacion>
</album>
```

Podemos observar que en general, un documento XML consta de:

1) Prólogo. Inicio del documento, describe qué tipo de documento es. Es similar al apartado head de HTML. Puede contener:

- Declaración del documento, que permite indicar el tipo de documento XML que es.
- Instrucciones para el procesamiento del documento.
- Comentarios.
- Indicación del esquema DTD, XSD o Relax NG para comprobar si el mismo es válido según las reglas impuestas por tales.
- Indicación de otros documentos que le afectan, como por ejemplo los documentos XSLT, los cuales permiten especificar la forma en la que el documento se debe mostrar en pantalla.

2) Ejemplar o elemento raíz. Contiene todo el contenido del documento y es único. Es un elemento obligatorio que se abre tras el prólogo y se debe cerrar justo al final. De este modo cualquier elemento está dentro del elemento raíz. Contiene:

- Más elementos, que pueden ser de diferentes tipos como: Padre, hermano, hijo o vacío.

- Atributos, añaden información a los elementos que acompañan, irán dentro de las etiquetas, separado del nombre por uno o más espacios.
- Entidades, tipo de marcado que identifica el nombre del archivo que queremos añadir al documento.
- Notaciones, tipo de marcado que establece la aplicación que procesa el archivo.
- Texto normal.
- Comentarios.

En cuanto a su sintaxis conviene citar una serie de consideraciones:

- XML, a diferencia de HTML, es "case sensitive", es decir, diferencia entre mayúsculas y minúsculas, de modo que, si un elemento se define como "Color", para referirnos a él si se escribe "color" dará error.

Ejemplo con Case sensitive y solapamiento.

```
<?xml version="1.0" ?>
<alumnos>
  <Nombre>Rober</nombre>
  <email>rober@ejemplo.com</email>
  <nombre>Claus</Nombre>
  <email>claus@ejemplo.com</email>
</alumnos>
```

- En la edición de documentos XML, debe cuidarse mucho la claridad del contenido, por lo que se suele organizar en varias líneas, que, normalmente, se componen de carácter de fin de línea y otro de retorno de carro. Un parser XML pasa únicamente a la aplicación el carácter de fin de línea.
- Debemos ser conscientes de que en la estructura de un documento XML, diferenciamos entre estructura lógica donde se recogen una serie de declaraciones, elementos, comentarios, etc. que se indican en el documento mediante marcas explícitas. Y, una estructura física donde figuran una serie de unidades llamadas entidades, es decir, son los que indican los datos que contiene el documento. Es muy importante la anidación correcta de las estructuras lógica y física.
- En la sintaxis, la importancia de los espacios en blanco, en la edición de documentos XML, merece que se preste especial atención a su utilización en el caso de "espacios en blanco" (espacios, tabuladores y líneas en blanco), considerados innecesarios para la versión que se sube a la red, y, entre el uso de

“espacios en blanco” que sí son representables y debe recogerse en el código fuente, por ejemplo.

Vamos a ver el ejemplo de `xml:space`, en el caso de aplicaciones que preserven la representación de esos espacios en blanco, el cual puede tomar los valores `preserve` o `default`, para conservar los espacios en blanco o aplicar el tratamiento por defecto, respectivamente. Un ejemplo sería:

```
<?xml version="1.0"?>
<refranes>
  <cita> Vísteme despacio que tengo prisa ?</cita>
  <popular xml:space="preserve">
    De los 40 para arriba,
    no te mojes la barriga
  </popular>
</refranes>
```

- En XML se denominan marcas a etiquetas, referencias de entidad y declaraciones, que son las partes del documento XML que entiende el parser, y el resto del documento contenido entre las marcas serán los datos que pueden interpretar las personas.

Las marcas en un documento XML son aquellas partes que empiezan con "<" y que acaban con ">" (por ejemplo, <Día>, o bien, en el caso de las referencias de entidad, empiezan por "&" y acaban con ";" (por ejemplo, " para representar las comillas dobles).



Sintaxis XML

Fuente: <https://proyectoatiblog.wordpress.com/unidad-4/>

- A la hora de utilizar nombres en XML, las reglas son que debe empezar con una letra y continúa con letras, dígitos, rayas, dos puntos o puntos, denominados de forma global como caracteres de nombre. Excepciones son los nombres que comiencen con las letras "xml", "XML" o "Xml", etc., se considerarán palabras reservadas y no podrán utilizarse.

- El lenguaje con el que se escribe el contenido, debe ser definido, mediante el atributo `xml:lang`, pudiendo insertarse tanto para especificar el lenguaje del contenido como de los valores de atributos que lleven los elementos.

En un documento XML, hemos visto que no existen etiquetas predeterminadas, sino que, deberemos crear unas etiquetas y darles el correspondiente significado.

En ello sí existen una serie de reglas generales de diseño que se deben cumplir, lo cual conferirá robustez al documento XML, como son:

- Debe respetarse la sintaxis de XML en sus documentos, lo que se conoce como documentos bien formados.
- Validez del documento XML, que se rige por el cumplimiento de una serie de reglas que serán definidas en el DTD o XML Schema, son de carácter opcional, pero sí recomendable.

Reglas a tener en cuentas tenemos:

- Sólo existirá un único elemento raíz.
- Es obligatorio el cierre en todos los elementos.
- Los elementos deben estar correctamente anidados.
- El entrecomillado en los valores de los atributos.
- Cumplimiento de las reglas propias en elementos y atributos.



ENLACE DE INTERÉS

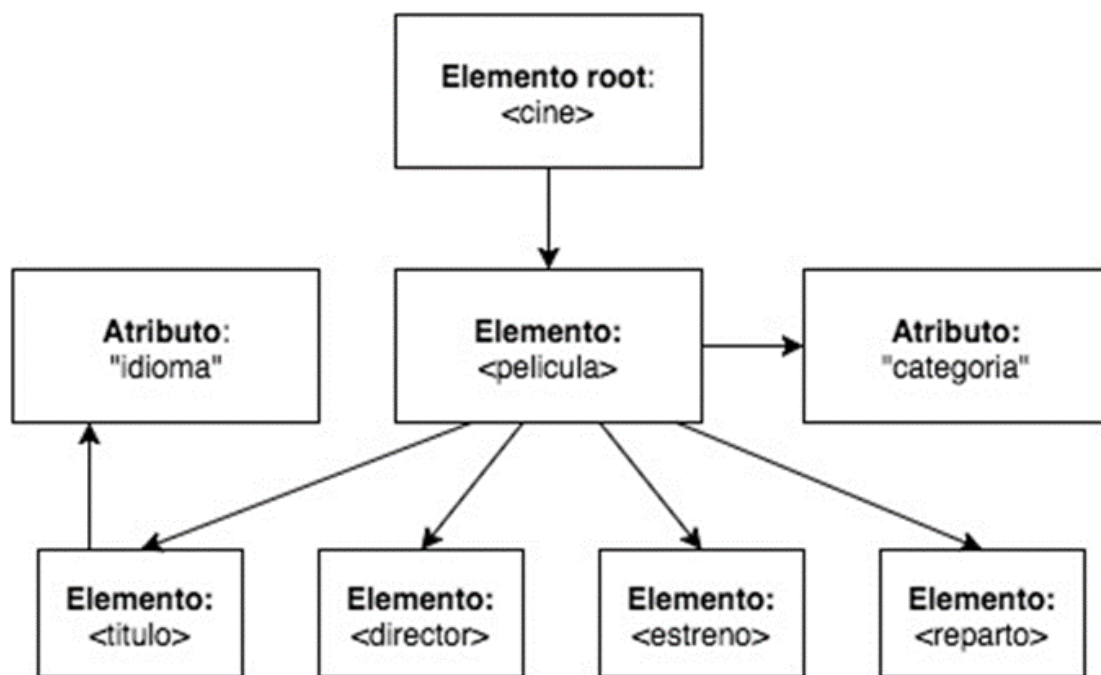
Accede a este enlace para ver un tutorial de XML:



2. CREACIÓN DE DESCRIPCIONES DE DOCUMENTOS, UTILIZACIÓN DE MÉTODOS DE DESCRIPCIÓN DE DOCUMENTOS XML

Una vez definida la estructura y conocida la sintaxis correcta a utilizar dentro de XML, debes ir añadiendo los distintos elementos dentro del documento con las correspondientes descripciones mediante la asignación de los atributos que los pueden acompañar y que conformarán las características propias del documento.

En la creación de documentos XML, tenemos las etapas de diseño, definición o declaración de su estructura y la de marcado de los documentos, en ellas se van consignando las descripciones del mismo a través de la declaración del tipo de documento y la definición del tipo de documento (DTD), donde se incluyen las declaraciones de: Tipos de elementos, listas de atributos, entidades y notaciones.



Ejemplo esquema XML

Fuente: <https://diego.com.es/xml-principios-basicos>

La declaración del tipo de elemento nos dice el nombre del tipo que seguirá un documento XML, en forma de enlace que figura en su elemento raíz, debiendo coincidir el nombre con el tipo, por ejemplo:

```
<!DOCTYPE MARCAS_COCHES SYSTEM "marcasCoches.dtd">
```

Donde se escribe en primer lugar la declaración del tipo <!DOCTYPE, y a continuación el nombre del tipo que es MARCAS_COCHES, para en último lugar pone el archivo donde se ha almacenado el documento XML con su contenido.

En la declaración del tipo de documento se define el tipo .dtd, que lleva asociadas las cualidades que posee, por lo que, debemos distinguir entre la declaración del tipo de documento como la que asigna el nombre al tipo, y la definición del tipo de documento, que es la que contiene la información sobre la sintaxis y restricciones a cumplir por ese tipo de documentos.

Importante que la definición del tipo de documento (DTD), se especifiquen los elementos, atributos, entidades y notaciones que puede utilizar el documento, así como si existe algún tipo de restricción estructural, de contenido, etc. Objetivo que se conseguirá mediante las correspondientes declaraciones de tipos de elementos, listas de atributos, entidades o notaciones, que puede ser de tipo externo en un documento distinto o internas, si se incluyen dentro del propio documento.

Declaraciones de tipo de elementos.

Van dirigidas a indicar al procesador XML cuáles son los elementos permitidos y qué contenido pueden tener, permitiendo la detección de errores, en el caso de incluir elementos no declarados, por ejemplo.

Su formato sería del tipo:

<!ELEMENT Nombre_XML_del_elemento contenido>

Los diferentes tipos de elementos son:

Elemento	Contenido
EMPTY	De tipo vacío.
ANY	Pueden contener caracteres y elementos sin restricciones.
Mixed	Contienen caracteres y elementos con restricciones.
Children	Sólo pueden contener otros elementos.

Declaraciones de listas de atributos.

Permiten definir el conjunto de atributos que va a tener cada elemento o los que puede contener, de manera que esta acción deberá realizarse con cada elemento creado.

Su formato sería del tipo:

```
<!ATTLIST nombre_elemento nombre_atributo tipo_atributo  
declaración_por_defecto>
```

Y en el caso de tener que incluir varios atributos para un mismo elemento, en vez de repetir varias veces la sentencia anterior, podemos agruparlos dentro de una sola instrucción que sería:

```
<!ATTLIST nombre_elemento nombre_atributo tipo_atributo declaración_por_defecto  
nombre_elemento nombre_atributo tipo_atributo declaración_por_defecto>
```

En la declaración debemos comenzar con <!ATTLIST, para a continuación ir incluyendo el nombre del elemento al que pertenece el atributo, separado por espacios. A continuación, y continuando las separaciones con espacios, deberemos ir incluyendo el tipo de atributo.

Deberemos tener en cuenta las opciones de declaración del atributo que podemos utilizar, en el ejemplo se ha utilizado “por defecto”, pero disponemos de las siguientes:

Declaración	Contenido del atributo
#REQUIRED	Necesita un valor, no se admite valor por defecto.
#IMPLIED	El contenido es opcional.
Valor por defecto	El atributo se inicializa a un valor concreto.
#FIXED	No cambian de valor, se mantienen en el documento.

Para finalizar, veremos cómo puede ser el contenido del atributo, limitando los valores que puede tomar mediante la especificación del tipo de atributo, donde destacamos los siguientes:

- CDATA o tipo cadena.

Admiten valores de cadenas de caracteres como pueden ser el en el caso de vehículos su tipo, como en el siguiente ejemplo:

```
<!ATTLIST VEHICULOS tipo CDTA #REQUIRED>
```

- Enumerado.

Se define una lista predeterminada de valores que permite el atributo, por ejemplo:

<!ATTLIST VEHICULOS tipo (turismo, SUV, familiar) #IMPLIED>

- ID.

Permite la identificación única de un elemento, por lo que no podrá existir otro elemento con un atributo tipo ID dentro del documento XML con el mismo valor. Un ejemplo lo tenemos en la siguiente instrucción:

<!ATTLIST VEHICULOS año ID #REQUIRED>

Con ella asignamos el atributo año de tipo ID, al elemento VEHICULOS.

Además, encontramos otros tipos de atributos con acciones diferentes y más complejas como son:

- ENTITY y ENTITIES, que se utilizan en entidades y notaciones para la inclusión de distintos tipos de objetos, por ejemplo, imágenes en el documento XML.
- NMTOKEN y NMTOKENS, con un funcionamiento similar a CDATA, pero en los que el valor sí debe cumplir las restricciones de los nombres XML, son poco utilizados.



RECUERDA

Las declaraciones de tipo de elementos indican al procesador XML qué elementos están permitidos y cuál es su contenido.

3. ASOCIACIÓN DE DESCRIPCIONES CON DOCUMENTOS. VALIDACIÓN

En este proceso de desarrollo web que estás, y una vez elaborado el documento XML, debes comprobar la corrección de la aplicación de reglas y sintaxis, mediante lo que se conoce como validación de documentos XML, que evitarán se produzcan errores en su ejecución.

A pesar de ser XML un lenguaje de sintaxis sencilla, dispone de un conjunto de herramientas que le confieren una gran potencialidad, sobre todo a la hora de una correcta asociación de descripciones en los documentos, gracias a una adecuada sintaxis y cumplimiento de reglas, permite una correcta estructura de los documentos y garantizar la consistencia tanto en contenido como en datos almacenados.

El primer paso, antes de su validación, será conseguir un documento bien formado, con una asociación de descripciones en sus elementos, atributos, etc., que cumpla las reglas de diseño, creando en su conjunto un lenguaje con características destacables como su robustez.

En esas reglas podemos destacar:

- Los documentos XML deben estar escritos con una sintaxis correcta, lo que se traduce en que el documento XML esté bien formado.
- Deben adecuarse a las reglas que se definan en el DTD o XML Schema, lo que le confiere validez.

En ese punto de documento bien formado, existen reglas que debe cumplir y que son:

1. El documento XML debe contar con un único elemento raíz.
2. Es obligatorio que todos los elementos estén cerrados.
3. Si se realizan anidaciones de elementos, deben estar correctamente estructuradas, en aperturas y cierres.
4. En los atributos, los valores deben ir entre comillas.
5. Son aplicables las reglas en nombres y atributos para una correcta escritura.

Y en lo que respecta a su validez, se conseguirá mediante la aplicación del DTD o XML Schema, donde estarán definidas las reglas semánticas.



XML DTD

Fuente: <https://www.educba.com/xml-dtd/>

DTD (Document Type Definition)

La validez de un documento XML, viene dada por, además de estar bien formado, cumplir con las reglas que recoja un determinado DTD.

El DTD contiene la definición del tipo de documento, en el sentido de recoger las reglas que sirven para su validación, no es obligatorio, pero si recomendable utilizarlo.

Encontramos dos maneras de asociar un DTD a un documento XML: Interna y externa.

- DTD interna.

Las reglas van definidas en el propio documento XML, a su favor, cuenta con simplificar el número de documentos y dejarlo en uno único, siendo quizá más compacto y sencillo el diseño.

En su redacción debe incluirse el elemento tipo DOCTYPE justo después de la declaración del documento XML, como en el siguiente ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE alumnos [
  <!ELEMENT alumnos (nuevos)*>
  <!ELEMENT nuevos (nombre, primerapellido, segundoapellido)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT primerapellido (#PCDATA)>
  <!ELEMENT segundoapellido (#PCDATA)>
]>
<alumnos>
  <nuevos>
    <nombre>Juan</nombre>
    <primerapellido>Blanco</primerapellido>
```



```
<segundoapellido>Peral</segundoapellido>
</nuevos>
<alumnos>
<nuevos>
  <nombre>Ana</nombre>
  <primerapellido>Vega</primerapellido>
  <segundoapellido>Luz</segundoapellido>
</nuevos>
<alumnos>
<nuevos>
  <nombre>Noelia</nombre>
  <primerapellido>Cano</primerapellido>
  <segundoapellido>Esera</segundoapellido>
</nuevos>
</alumnos>
```

- DTD externa.

En ellos los elementos se declaran fuera de un documento XML, pudiendo acceder a ellos al especificar los atributos del sistema que pueden consistir en un archivo tipo .dtd o una dirección URL, debiendo asignar al atributo standalone el valor no, para que el procesador XML interprete que existe un DTD externa.

Podemos distinguir entre:

1) DTD externa privada

Donde la sintaxis sería:

```
<!DOCTYPE elemento-raíz SYSTEM "URI">
Y a modo de ejemplo tendríamos el siguiente código:
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE alumnos [
  <!ELEMENT alumnos (nuevos)*>
  <!ELEMENT nuevos (nombre, primerapellido, segundoapellido)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT primerapellido (#PCDATA)>
  <!ELEMENT segundoapellido (#PCDATA)>
]>
```

Encontrando el siguiente documento XML válido

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE alumnos SYSTEM "alumnos.dtd">
```

```
<alumnos>
  <nuevos>
    <nombre>Juan</nombre>
    <primerapellido>Blanco</primerapellido>
    <segundoapellido>Peral</segundoapellido>
  </nuevos>
</alumnos>
<alumnos>
  <nuevos>
    <nombre>Ana</nombre>
    <primerapellido>Vega</primerapellido>
    <segundoapellido>Luz</segundoapellido>
  </nuevos>
</alumnos>
<alumnos>
  <nuevos>
    <nombre>Noelia</nombre>
    <primerapellido>Cano</primerapellido>
    <segundoapellido>Esera</segundoapellido>
  </nuevos>
</alumnos>
```

En este documento XML, haciendo uso de una DTD externa privada, se ha escrito una lista de marcadores de páginas web, guardando de cada uno de ellos su nombre, una descripción y su URL.

2) DTD externa pública

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <title>Título</title>
  </head>
  <body>
    <p>Párrafo</p>
  </body>
</html>
```

En este caso, se utiliza un Identificador Público Formal (FPI-Formal Public Identifier), concretamente: //W3C//DTD XHTML 1.0 Strict//EN, desde la url de w3.org, para identificar una entrada en el catálogo.



EJEMPLO PRÁCTICO

Roberto forma parte del equipo de desarrollo web del departamento de informática de su empresa, dentro del proyecto de creación de un sitio web para una academia de idiomas.

Dada la importancia del manejo correcto del lenguaje XML y su aplicación en el desarrollo web, es necesaria la creación de un DTD que contenga la definición del tipo de documento, y recoja las reglas que sirven para su validación.

Documento que deberá crear para este caso, con el fin de facilitar la gestión de alumnos de la academia, en concreto quieren controlar los nuevos alumnos que se están matriculando para gestionar la oferta de bienvenida que tenía publicitada. ¿Cómo sería el código fuente para la creación del DTD?

Solución.

Y a modo de ejemplo tendríamos el siguiente código:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE alumnos [
  <!ELEMENT alumnos (nuevos)*>
  <!ELEMENT nuevos (nombre, primerapellido, segundoapellido)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT primerapellido (#PCDATA)>
  <!ELEMENT segundoapellido (#PCDATA)>
]>
```



VÍDEO DE INTERÉS

En este vídeo se explica cómo validar con DTD externo un documento XML:



4. HERRAMIENTAS DE CREACIÓN Y VALIDACIÓN

En esta parte del desarrollo web, además de ese punto creativo de los desarrolladores, existe una labor de escritura de código, que suele ser quizá el proceso más tedioso. Para facilitar esa tarea deberás utilizar algún tipo de herramienta que permita la creación y validación de los documentos XML, con la finalidad de minimizar los esfuerzos a la hora de esa escritura de código fuente.

En un principio, un documento XML está formado por texto plano, por lo que si hablamos de herramientas para su creación o edición, un simple editor de texto sería suficiente para ello, pero resulta evidente que la simplicidad de un editor de texto, carente de cualquier tipo de ayuda a nivel sintáctico, de códigos de color, etc., las convierte en herramientas no muy recomendables, siendo más habitual el uso de herramientas que sí prestan diferentes tipos de ayuda a la hora de crear un documento XML.

En el tema de validación, es un proceso sobre un documento XML en el que es imprescindible la utilización de una herramienta eficaz que permite la validación con un DTD o un XML Schema.

Comenzaremos por las herramientas para creación, edición o modificación de un documento XML, donde encontramos:

1) Editores de texto

Son programas compatibles con los distintos sistemas operativos que van desde el básico Bloc de Notas que ofrece Windows a otras herramientas más complejas como Visual Studio Code o Notepad ++, entre otras.

En el nivel más bajo encontramos las que no ofrecen ningún tipo de ayuda en la escritura de código ni de manejo de XML, como es el caso del Bloc de Notas. Avanzando en complejidad, encontramos ya editores de texto que sí permiten reconocer la gramática y sintaxis propia de XML, ofreciendo ayudas incluso de tipo visual coloreando los textos en función de lo que representan (elementos, atributos, etc.), incluso incorporando la función acordeón para desplegar ciertos elementos y consultar su contenido.

Veremos, a continuación, una breve descripción de una de ellas, Visual Studio Code.



Visual Studio Code

Visual Studio Code

Fuente: <https://www.stickpng.com/es/img/dibujos-animados/logotipos-de-editoriales/logotipo-completo-de-visual-studio-code>

Es un editor libre de código fuente desarrollado por Microsoft, en el que destacan características como:

- Multiplataforma (Windows, GNU/Linux, macOS).
- Integración con Git.
- Autocompletado y resaltado de sintaxis (IntelliSense).
- Soporte para depuración de código para detección de errores.
- Gran variedad de extensiones.
- Uso de control de versiones.
- Admite varios lenguajes de programación.
- Gran popularidad y comunidad de usuarios.



ENLACE DE INTERÉS

Tienes disponible su descarga desde la web:



2) Navegadores web

Presentan los documentos XML en forma de árbol que permite expandir o comprimir sus ramas, para una mejor comprensión de su estructura. Añaden ayudas visuales como diferenciación del código por colores.

Un ejemplo sería abrir un documento XML en Edge, pinchado sobre el archivo XML con el botón derecho y seleccionando Edge en Abrir con, devolviendo una pantalla similar a la que podemos ver en la siguiente imagen.

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Comprobante xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="3.3" Serie="MP" Fecha="2020-10-02
000092 Ref_Vta:000091" SubTotal="26531.08" Descuento="0.00" Moneda="MXN" TipoCambio="1" Total="30776.05" Tip
xsi:noNamespaceSchemaLocation="https://facturador.zendesk.com/hc/es/articles/115002595691-XSD-Emission-3-3/XML
  <CfdiRelacionados TipoRelacion="01">
    <CfdiRelacionado UUID="86277CAE-BED7-4A80-A03F-6E7E606D79BA"/>
    <CfdiRelacionado UUID="3F97A046-9887-401F-91CC-5C2B2C3807DD"/>
    <CfdiRelacionado UUID="65D4BA0E-27C8-430B-83C3-5E4185834038"/>
  </CfdiRelacionados>
  <Emisor Rfc="GOYA780416GM0" RegimenFiscal="612"/>
  <Receptor Rfc="LAN7008173R5" Nombre="VENTA AL PUBLICO EN GENERAL" UsoCFDI="G02">
    <Domicilio Calle="BLVD. MANUEL AVILA CAMACHO #6A ESQ.JOSELILLO P-118" Colonia="LOMAS DE SOTELO" Localidad
  </Receptor>
  <Conceptos>
    <Concepto ClaveProdServ="43191600" NoIdentificacion="TEL CEL MOT G8POL 64GB" Cantidad="1" ClaveUnidad="H8
ValorUnitario="3412.97" Importe="3412.97" Descuento="0">
      <Impuestos>
        <Impuesto TipoImpuesto="Traslado" Base="3412.97" Impuesto="002" TipoFactor="Tasa" TasaOCuota="0.160000
      </Impuestos>
    </Concepto>
    <Concepto ClaveProdServ="43191501" NoIdentificacion="TEL CEL SMG S10 128" Cantidad="1" ClaveUnidad="H87"
ValorUnitario="13561.21" Importe="13561.21" Descuento="0">
      <Impuestos>
        <Impuesto TipoImpuesto="Traslado" Base="13561.21" Impuesto="002" TipoFactor="Tasa" TasaOCuota="0.160000
      </Impuestos>
    </Concepto>
    <Concepto ClaveProdServ="43191600" NoIdentificacion="TEL CEL XMI PF2P 128GB" Cantidad="1" ClaveUnidad="H8
ValorUnitario="9556.90" Importe="9556.90" Descuento="0">
      <Impuestos>
        <Impuesto TipoImpuesto="Traslado" Base="9556.90" Impuesto="002" TipoFactor="Tasa" TasaOCuota="0.160000
      </Impuestos>
    </Concepto>
  </Conceptos>
  <AddendaFE TipoCFD="Nota de Credito"/>
</Comprobante>
```

Documento XML en Edge

Fuente: <https://helpdesk.facturador.com/hc/es/articles/115002599151-ARCHIVOS-DE-PRUEBA-XML>

3) Editores web

Son la opción de disponer de herramientas para creación de documentos XML en modo online, a través de un navegador web.

Ofrecen funcionalidades como el sangrado automático, representación en forma de árbol, codificación con colores, etc. Además de comprobación de sintaxis y validación con DTD y XML Schema.

Algunos ejemplos son XML Editor o XML Viewer, por ejemplo, del que veremos sus características a continuación.



XML Viewer

XML Viewer

Fuente: <https://chrome.google.com/webstore/detail/xml-viewer/legopflakafagikcpapgnbokgkbejlk?hl=es-419>

Editor de código de carácter gratuito, desarrollado por MindFusion para Windows, que destaca por:

- Árbol personalizado de colores.
- Opciones de personalización.

- Permite envolver los nodos XML.
- Creación de marcadores y añadir números de línea.
- Renderiza archivos SVG.
- Carga múltiple de archivos XML.



ENLACE DE INTERÉS

Tienes disponible su descarga en diferentes páginas de descarga de software, o, como extensión de Google Chrome desde:



4) Eclipse

Es un popular IDE (Integrated Development Environment) o entorno de desarrollo integrado, que permite la programación en distintos lenguajes, incluyendo extensiones para la gestión de documentos XML.



Eclipse

Fuente: <https://ahorasomos.izertis.com/globetesting/2019/03/alm-octane-pensando-en-integraciones/logoeclipse2/>

En concreto cuenta con Eclipse XML Editors and Tools, como herramienta muy completa en la edición de documentos XML.

Destaca por aspectos como:

- Optimización de entono de trabajo mediante ventanas y editores preconfigurados.
- Relaciona recursos entre sí para gestión de proyectos.

- Depurador de código.
- Amplia variedad de plugins, gratuitos y de pago.
- Reconocimiento sintáctico de código a través de colores.

Descargable desde su página web: <https://www.eclipse.org/downloads/>



EJEMPLO PRÁCTICO

Noelia, debido a su alta cualificación en la programación con lenguajes de marcas, es la responsable dentro del departamento de informática de su empresa, de la elección de cualquier nueva herramienta o software que se vaya a incorporar en ella.

Ante la gran oferta existente en el mercado de herramientas que se emplean en el desarrollo web, los compañeros que se trabajan en estas tareas, le han solicitado que realice una evaluación de las herramientas de edición y validación de código más utilizadas y con mayor comunidad de usuarios.

El objetivo es minimizar la carga de trabajo del departamento de desarrollo en lo relacionado a la edición y validación de código con la utilización de alguna herramienta de este tipo, que se considere óptima y rentable.

¿Qué dos herramientas podría destacar y qué ventajas aportarían?

Solución.

En la actualidad, dos de las herramientas a destacar serían las siguientes:

Visual Studio Code

Es un editor libre de código fuente desarrollado por Microsoft, en el que destacan características como:

- Multiplataforma (Windows, GNU/Linux, macOS).
- Integración con Git.
- Autocompletado y resaltado de sintaxis (IntelliSense).
- Soporte para depuración de código para detección de errores.
- Gran variedad de extensiones.
- Uso de control de versiones.
- Admite varios lenguajes de programación.
- Gran popularidad y comunidad de usuarios.

Eclipse

En concreto cuenta con Eclipse XML Editors and Tools, como herramienta muy completa en la edición de documentos XML.

Destaca por aspectos como:

- Optimización de entorno de trabajo mediante ventanas y editores preconfigurados.
- Relaciona recursos entre sí para gestión de proyectos.
- Depurador de código.
- Amplia variedad de plugins, gratuitos y de pago.
- Reconocimiento sintáctico de código a través de colores.



VÍDEO DE INTERÉS

Conoce cómo visualizar un archivo XML en Windows en este vídeo:



RESUMEN FINAL

En esta unidad hemos visto la importancia de la definición de esquemas (estructura) y vocabulario (sintaxis) en la creación de documentos mediante lenguaje de marcas en entornos web.

Con respecto a las tecnologías para la definición de documentos, nos hemos centrado en concreto en XML, por ofrecer muy diversas tecnologías como son XSL, XPath, XLink, XPointer o XQL.

Parte importante es también conocer la estructura de los documentos XML, que, a pesar de su similitud con los creados con HTML, conviene diferenciar y conocer cómo XML sí indica los datos que se describen, tanto en su parte inicial o prólogo (el tipo de documento) como en su elemento raíz o ejemplar, que recoge todo el contenido restante en forma de: Elementos, atributos, entidades, notaciones o comentarios.

Precisamente ese contenido que se detalla en el documento XML es posible adaptarlo a nuestras necesidades mediante las correspondientes descripciones, como pueden ser los atributos que acompañan a los elementos, a través de un DTD, que puede ser interno o externo.

En la comprobación del funcionamiento de un documento XML cuando es procesado por un navegador web, sí es muy importante que esté bien formado, además de que su elaboración haya sido de acuerdo a las reglas establecidas por el propio XML, bien sea a través del DTD o del XML Schema, que le va a dotar de un alto grado de robustez.

Para finalizar la unidad, se ha hecho un recorrido por las diferentes herramientas de creación y validación de documentos XML que podemos encontrar en el mercado, desde las más sencillas como son los editores de texto tipo Visual Studio Code o Notepad++, pasando por otras un poco más avanzadas que nos ofrecen los propios navegadores web, directamente como Edge o Chrome, para continuar con ya editores webs propiamente dichos, entre los que destacamos XML Viewer. Poniendo el punto final con la herramienta Eclipse, un popular IDE o entorno de desarrollo integrado, compatible, además de XML, con otro gran número de lenguajes de programación.