



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



### XML

- XML significa lenguaje de marcado extensible
- XML fue diseñado para almacenar y transportar datos
- XML fue diseñado para ser legible tanto por humanos como por máquinas
- XML es una recomendación del W3C
- XML no hace nada

<nota>

<para>Mis alumnos</para>

<de>Patty</de>

<título>Recordatorio</título>

<mensaje>No olviden estudiar este fin de semana largo</mensaje>

</nota>

El ejemplo es muy descriptivo:

Tiene información del remitente y del receptor,  
Tiene un título, y  
Tiene un mensaje

Pero no hace nada

Es solo información envuelta en etiquetas.

Alguien deberá escribir un software para enviarlo, recibirlo, almacenarlo o mostrarlo.

#### Nota

Para: Mis alumnos

De: Patty

#### Recordatorio

No olviden estudiar este fin de semana largo

### XML y HTML fueron diseñados con diferentes objetivos:

- XML fue diseñado para transportar datos, enfocado en qué datos son
- HTML fue diseñado para mostrar datos, enfocado a cómo se ven los datos
- Las etiquetas XML no están predefinidas como las etiquetas HTML

Las etiquetas en XML son inventadas por el autor de documento XML.

Con XML, el autor deberá definir la estructura del documento y las etiquetas.



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Mercado Extensible (XML)



### XML es Extensible

La mayoría de las aplicaciones XML funcionarían como se espera incluso si se agregan (o eliminan) nuevos datos.

Imagine una aplicación diseñada para mostrar la versión original de note.xml (<para><de><titulo><mensaje>).

Luego imagine una versión más nueva de note.xml con elementos <date> y <hour> agregados, y un <titulo> eliminado.

La forma XML es construida, la versión anterior de la aplicación aún puede funcionar:

```
<nota>
  <date>2020-03-13</date>
  <hour>12:30</hour>
  <para>Mis alumnos</para>
  <de>Patty</de>
  <mensaje>No olviden estudiar este fin de semana largo</mensaje>
</nota>
```

Nota	Nota
Para: Mis alumnos	Para: Mis alumnos
De: Patty	De: Patty
Recordatorio	Date: 2020-03-13 12:30
No olviden estudiar este fin de semana largo	No olviden estudiar este fin de semana largo

### XML simplifica las cosas

- Simplifica el intercambio de datos
- Simplifica el transporte de datos
- Simplifica los cambios de plataforma
- Simplifica la disponibilidad de datos

Muchos sistemas informáticos contienen datos en formatos incompatibles. El intercambio de datos entre sistemas incompatibles (o sistemas actualizados) es una tarea que requiere mucho tiempo para los desarrolladores web. Se deben convertir grandes cantidades de datos y, a menudo, se pierden datos incompatibles.

XML almacena datos en formato de texto plano. Esto proporciona una forma independiente de software y hardware de almacenar, transportar y compartir datos.



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Mercado Extensible (XML)



XML también facilita la expansión o actualización a nuevos sistemas operativos, nuevas aplicaciones o nuevos navegadores, sin perder datos.

Con XML, los datos pueden estar disponibles para todo tipo de "máquinas de lectura" como personas, computadoras, máquinas de voz, noticias, etc.

### **XML es una recomendación de la W3C desde febrero de 1998.**

XML se usa a menudo para separar los datos de la presentación.

XML no contiene ninguna información sobre cómo mostrarse.

Los mismo datos XML se pueden usar en muchos escenarios de presentación diferentes, debido a esto, con XML existe la separación total entre los datos y la presentación.

XML se usa para almacenar o transportar datos, mientras que HTML se usa para formatear y mostrar los mismos datos.

Al mostrar datos en HTML, no deberíamos tener que editar el archivo HTML cuando los datos cambian.

Con pocas líneas de código JavaScript, puede leer un archivo XML y actualizar el contenido de datos de cualquier página HTML.

### **Ejemplo de la librería**

Title	Author
Everyday Italian	Giada De Laurentiis
Harry Potter	J K. Rowling
XQuery Kick Start	James McGovern
Learning XML	Erik T. Ray

Existen miles de formatos XML, en industrias diferentes, para describir las transacciones diarias de datos:

- Transacciones financieras
- Datos Médicos
- Datos matemáticos
- Mediciones científicas
- Información de noticias
- Servicios meteorológicos



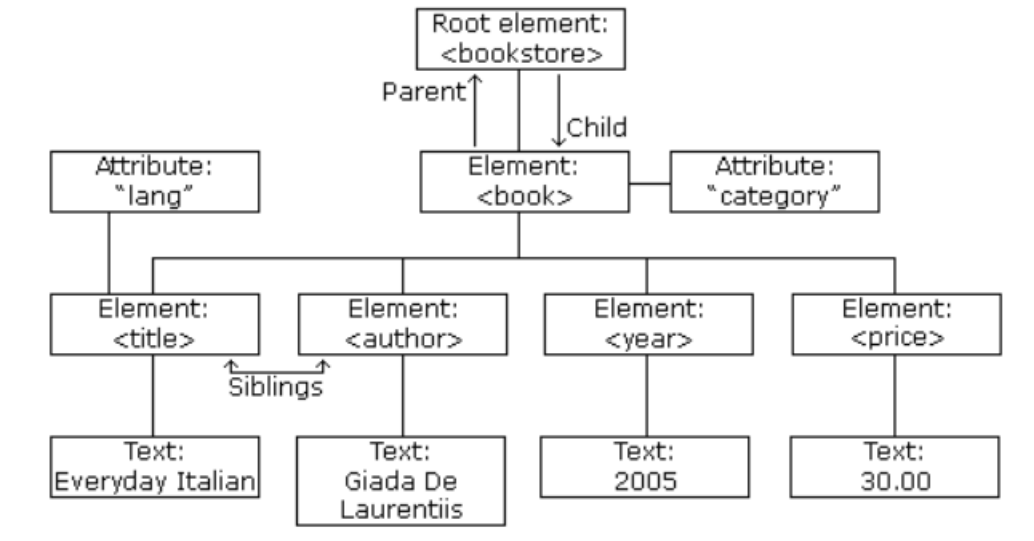
# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



### 3.1 Elementos de XML

Los documentos XML forman una estructura de árbol que comienza en “la raíz” y se ramifica a “las hojas”.



Los documentos XML son formados como árboles de elementos.

Un árbol XML inicia con el elemento root y sus ramas son elementos child.

Todos los elementos pueden tener sub elementos.

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

Los términos parent, child y sibling son usados para describir las relaciones entre los elementos.

### Sintaxis

XML utiliza una sintaxis muy autodescriptiva.

**<?xml versión="1.0" encoding="UTF-8"?>**

EL prólogo es opcional, pero si se pone debe de ser al inicio del documento.



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Mercado Extensible (XML)



La primera línea es el elemento root del documento:

<libreria>

La siguiente línea comienza con un elemento:

<book>

El elemento <book> tiene 4 elementos: <title>, <autor>, <year> y <Price>

Y después se coloca la línea final del elemento </book>

Un **elemento** puede contener:

- Texto
- Atributos
- Otros elementos
- O una combinación

**Elementos vacíos:** <element></element> ó <element /> Create descriptive names, like this: <person>, <firstname>, <lastname>.

### Reglas de nomenclatura XML

Los elementos XML deben seguir estas reglas de nomenclatura:

- Los nombres de los elementos distinguen entre mayúsculas y minúsculas
- Los nombres de los elementos deben comenzar con una letra o un guión bajo
- Los nombres de elementos no pueden comenzar con las letras xml (o XML, o Xml, etc.)
- Los nombres de elementos pueden contener letras, dígitos, guiones, guiones bajos y puntos.
- Los nombres de elementos no pueden contener espacios
- Se puede usar cualquier nombre, no se reservan palabras (excepto xml)

### Mejores prácticas de nomenclatura

Cree nombres descriptivos como: <person>, <firstname>, <lastname>.

Cree nombres simples y cortos como: <book\_title> no <the\_title\_of\_the\_book>.

Evite "-". Por ejemplo: si nombra algo como "first-name", algunos programas pueden pensar que desea obtener "name" de "first".



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



Evite ".". Por ejemplo: si nombra algo como "first.name", algunos programas pueden pensar que "name" es una propiedad del objeto "first".

Evite ":". Ya que están reservados para espacios de nombres.

### Estilos de nombres

No hay un estilo de nombres definido para elementos XML, pero los siguientes son los usados comúnmente:

Style	Example	Description
Lower case	<firstname>	All letters lower case
Upper case	<FIRSTNAME>	All letters upper case
Underscore	<first_name>	Underscore separates words
Pascal case	<FirstName>	Uppercase first letter in each word
Camel case	<firstName>	Uppercase first letter in each word except the first

Los documentos XML tienen frecuentemente una correspondiente base de datos. Una práctica común es usar las reglas de nomenclatura de la base de datos para los elementos XML.

Como se mencionó anteriormente todos los elementos XML pueden tener atributos, con el HTML. Éstos son diseñados para contener información relacionada a un específico elemento. Los atributos siempre deben estar entre comillas, por ejemplo:

**<person gender="female">**

Si el valor del atributo contiene comillas, puedes usar comillas simples, como esto:

**<gangster name='George "Shotgun" Ziegler'>**

Ó utilizar los caracteres: **<gangster name="George &quot;Shotgun&quot; Ziegler">**

Los atributos también pueden ser representados como elementos, no hay reglas acerca de cuándo usar atributos o cuando elementos en XML.



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)

### Atributos XML para metadatos



Algunas veces las referencias ID son asignadas a los elementos, es decir para identificar elementos XML de la misma forma que el atributo id en HTML. En el siguiente código, podemos observar que el atributo ID, nos permite identificar diferentes notas:

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

Es decir, los metadatos (datos sobre datos) deben almacenarse como atributos, y los datos en sí como elementos.

### Conflicto de nombres

Debido a que los nombres de los elementos son definidos por los desarrolladores, cuando mezclamos varios documentos XML de diferentes aplicaciones XML, frecuentemente se genera un conflicto, es por ello que para resolver esto debemos utilizar un prefijo de nombre:

```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>

<f:table>
  <f:name>African Coffee Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



Para usar prefijos en XML deben ser definidos por un **namespace**. El namespace puede ser definido como un atributo **xmlns** en la etiqueta de inicio de un elemento, teniendo la siguiente sintaxis: **xmlns:prefijo="URI"** (por ejemplo: **xmlns:h="http://www.w3.org/TR/html4/"**) o en el elemento root del XML: **<root xmlns:h=http://www.w3.org/TR/html4/>**

El propósito de usar un URI es para dar un único nombre al namespace, no para buscar información.

Un **Identificador Uniforme de Recursos (URI)** es una cadena de caracteres que identifica un recurso de Internet.

El URI más común es el Localizador Uniforme de Recursos (URL) que identifica una dirección de dominio de Internet.

### 3.2 Definición del tipo de documento (DTD)

Un DTD define la estructura y los elementos y atributos legales de un documento XML.

En un DTD, los elementos son declarados con la siguiente sintaxis:

```
<!ELEMENT element-name category>  
or  
<!ELEMENT element-name (element-content)>
```

Un documento XML "válido" está "bien formado", si cumple con las reglas de un DTD:

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE note SYSTEM "Note.dtd">  
<note>  
<to>Tove</to>  
<from>Jani</from>  
<heading>Reminder</heading>  
<body>Don't forget me this weekend!</body>  
</note>
```

La declaración DOCTYPE contiene una referencia a un archivo DTD. El contenido del DTD, es el siguiente:





# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



Note.dtd:

```
<!DOCTYPE note
[
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
```

Éste es interpretado así:

- ✓ **!DOCTYPE note** – Define que el elemento root del documento es nota
- ✓ **!ELEMENT note** – Define que el elemento note, debe contener los elementos: "to, from, heading body"
- ✓ **!ELEMENT to** – Define que el elemento to, sea de tipo "#PCDATA"
- ✓ **!ELEMENT from** – Define que el elemento from, sea de tipo "#PCDATA"
- ✓ **!ELEMENT heading** – Define que el elemento heading, sea de tipo "#PCDATA"
- ✓ **!ELEMENT body** – Define que el elemento body, sea de tipo "#PCDATA"

El tipo **#PCDATA**, es un dato caracter que se puede analizar. Como por ejemplo, el texto que se encuentra entre la etiqueta de inicio y la etiqueta final de un elemento XML. Un analizador examinará ese texto en busca de entidades y marcas. Las etiquetas dentro del texto se tratarán como marcas y las entidades se expandirán.

**CDATA** significa datos de caracteres. Es texto que no será analizado. Las etiquetas dentro del texto NO serán tratadas como marcas y las entidades no serán expandidas.

También podemos declarar un DTD dentro de un archivo XML, éste deberá estar dentro de la definición **<!DOCTYPE>**, de la siguiente forma:

```
<?xml version="1.0"?>
<!DOCTYPE note [
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
]>
<note>
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this weekend</body>
</note>
```



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Mercado Extensible (XML)



### Uso de DTD para declaración de entidad

Una declaración DOCTYPE puede ser usada para definir caracteres especiales o cadenas, usadas en el documento y ésta tiene 3 partes: comienza con un **&**, luego viene el **nombre de la entidad** y termina con un **;**.

Lo anterior lo podemos visualizar en el siguiente ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE note [
  <!ENTITY nbsp "&#xA0;">
  <!ENTITY writer "Writer: Donald Duck.">
  <!ENTITY copyright "Copyright: W3Schools.">
]>

<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
  <footer>&writer;&nbsp;&copyright;</footer>
</note>
```

```
▼ <note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
  <footer>Writer: Donald Duck. Copyright: W3Schools.</footer>
</note>
```

Con un DTD puedes verificar que los datos que recibes del mundo exterior sean válidos, también puedes usar un DTD para verificar tus propios datos, se puede acordar entre un grupo de personas usar un DTD estándar para intercambiar datos.

XML no requiere un DTD, cuando trabaje con archivos XML pequeños, crear una DTD puede ser una pérdida de tiempo.

Si se desarrollan aplicaciones, espera hasta que la especificación sea estable antes de agregar un DTD, sino el software podría dejar de funcionar debido a errores de validación.



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



### 3.3 Esquema de documentos XML (XML Schema)

Un esquema XML describe la estructura de un documento XML, justo como un DTD.

El lenguaje de esquema XML es también se conoce como definición de esquema XML (XSD).

El Esquema XML es una alternativa basada en XML a DTD:

```
<xs:element name="note">

  <xs:complexType>
    <xs:sequence>
      <xs:element name="to" type="xs:string"/>
      <xs:element name="from" type="xs:string"/>
      <xs:element name="heading" type="xs:string"/>
      <xs:element name="body" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>

</xs:element>
```

Y el cual es interpretado así:

- ✓ **<xs:element name="note">** - Define el elemento llamado "note"
- ✓ **<xs:complexType>** - El elemento "note" es de tipo complejo
- ✓ **<xs:sequence>** - El tipo complejo es una secuencia de elementos
- ✓ **<xs:element name="to" type="xs:string">** - El elemento "to" es de tipo cadena (texto)
- ✓ **<xs:element name="from" type="xs:string">** - El elemento "from" es de tipo cadena
- ✓ **<xs:element name="heading" type="xs:string">** - El elemento "heading" es de tipo cadena
- ✓ **<xs:element name="body" type="xs:string">** - El elemento "body" es de tipo cadena

El propósito de un esquema XML es definir los bloques de construcción legales de un documento XML:

- ✓ Los elementos y atributos pueden aparecer en un documento
- ✓ El número de (y el orden de) elementos child



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Marcado Extensible (XML)



- ✓ Tipos de datos para elementos y atributos
- ✓ Valores predeterminados y fijos para elementos y atributos

En el mundo XML, cientos de formatos XML estandarizados son diariamente usados. Muchos de esos estándares están definidos por esquemas XML.

Los esquemas XML son más potentes que los DTD:

- ✓ **Son escritos en XML:**
  - No tienes que aprender un nuevo lenguaje
  - Puedes utilizar su editor XML para editar sus archivos esquema
  - Puedes usar su analizador para analizar sus archivos esquema
  - Puedes manipular tus esquemas con XML DOM
  - Puedes transformar tus esquemas con XSLT
- ✓ **Son extensibles a las adiciones,** porque están escritos en XML:
  - Reutilice su esquema en otros esquemas
  - Cree sus propios tipos de datos derivados de los tipos estándar
  - Referencie múltiples esquemas en el mismo documento
- ✓ **Soportan tipos de datos** (Una de las mayores fortalezas de los esquemas XML es la compatibilidad con los tipos de datos):
  - Es más fácil describir el contenido del documento permitido
  - Es más fácil definir restricciones en los datos
  - Es más fácil validar la exactitud de los datos
  - Es más fácil definir patrones de datos
  - Es más fácil convertir datos entre diferentes tipos de datos
- ✓ **Soportan namespaces**

### ¿Por qué usar un esquema XML?

Con esquema XML, los archivos XML pueden llevar una descripción de su propio formato, se puede acordar un estándar para intercambiar datos y se pueden verificar los datos.

### Comunicación Segura de Datos

Cuando se envían datos de un remitente a un receptor, es esencial que ambas partes tengan las mismas "expectativas" sobre el contenido.

Con los esquemas XML, el remitente puede describir los datos de manera que el receptor los entienda.

Una fecha como: "03-11-2004", en algunos países, se interpretará como 3.Noviembre y en otros países como 11.Marzo.

Sin embargo, un elemento XML con un tipo de datos como este:



# U.A. Tecnologías para la Web

## Unidad Temática III. Lenguaje de Mercado Extensible (XML)



`<date type = "date"> 2004-03-11 </date>`

garantiza una comprensión mutua del contenido, porque el tipo de datos XML "fecha" requiere el formato "AAAA-MM-DD".

Para referenciar un esquema en un documento XML, se debe hacer de la siguiente forma:

```
<?xml version="1.0"?>

<note xmlns="https://www.w3schools.com"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://www.w3schools.com note.xsd">

  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

`xmlns="https://www.w3schools.com"` especifica la declaración de espacio de nombres predeterminada. Esta declaración le dice al validador de esquema que todos los elementos utilizados en este documento XML se declaran en el espacio de nombres "https://www.w3schools.com".

Una vez que tenga disponible el espacio de nombres de instancia de esquema XML:

```
xmlns: xsi = "http://www.w3.org/2001/XMLSchema-instance"
```

Tu puedes usar el atributo `schemaLocation`. Este atributo tiene dos valores, separados por un espacio. El primer valor es el espacio de nombres a utilizar. El segundo valor es la ubicación del esquema XML que se utilizará para ese espacio de nombres:

```
xsi:schemaLocation="https://www.w3schools.com note.xsd"
```