

# Ciclo Formativo de Grado Superior de Administración de Sistemas Informáticos en red



Módulo Profesional: **LMSGI**

Unidad de Trabajo 3.- Introducción al Lenguaje de Marcas – XML (Extensible Markup Language)

*Departamento de Informática y Comunicación  
IES San Juan Bosco (Lorca-Murcia)  
Profesor: Juan Antonio López Quesada*



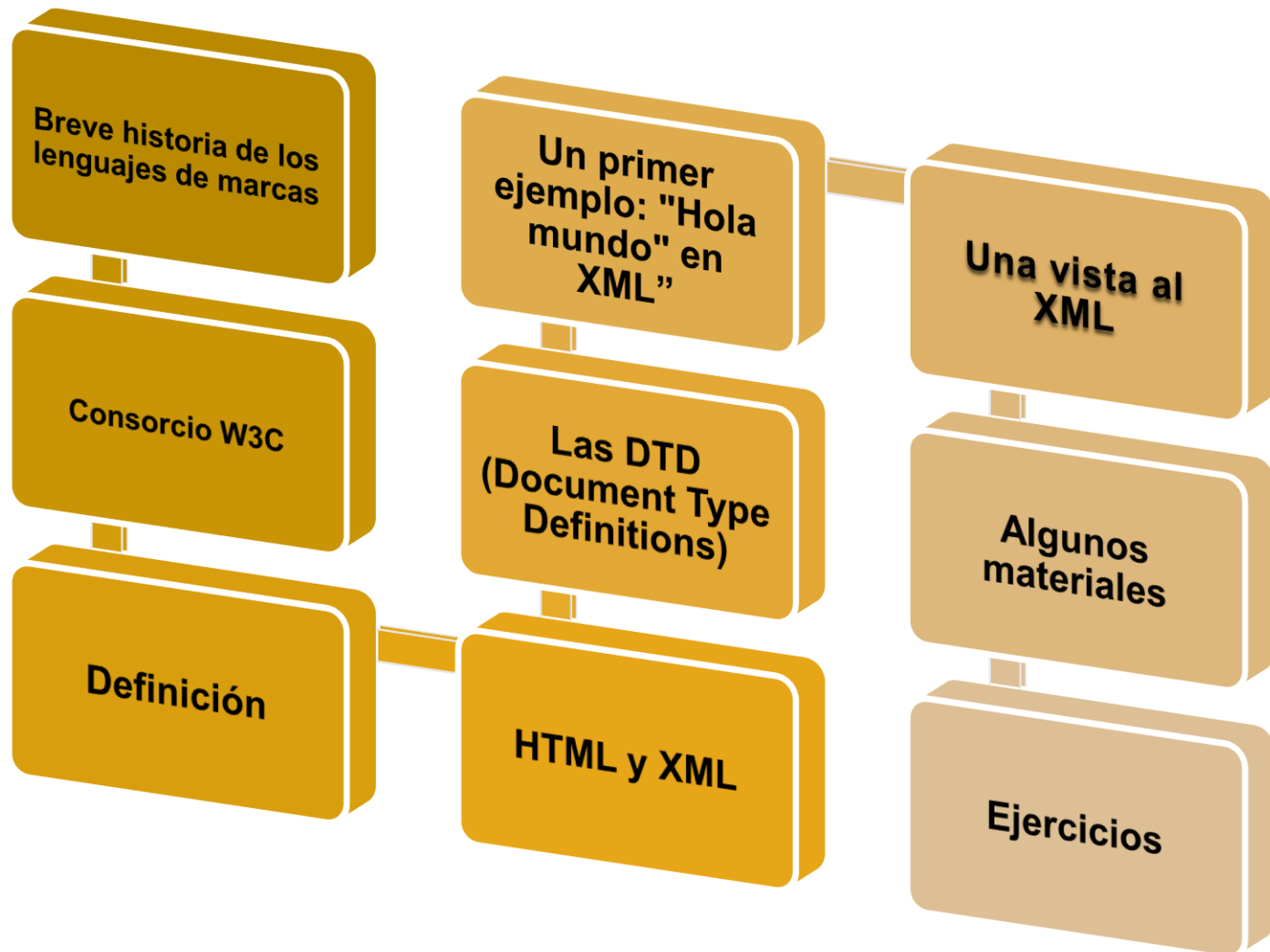
# Abstract/Resumen

**XML (Extensible Markup Language, lenguaje de marcas extensible)** es la última palabra de moda en Internet. También es una tecnología que madura rápidamente con múltiples aplicaciones en el mundo real, particularmente para la gestión, visualización y organización de los datos. En conjunto con su lenguaje de visualización **XSL** y el modelo de objeto documento **DOM**, es una tecnología esencial para cualquier usuario de lenguajes de marcas en la red o en la intranet.

*El objetivo de este tema es iniciar al alumno en los conceptos básicos de XML. Conocer los elementos que componen un documentos XML. Conocer el organismo W3C, consorcio que emite las diferentes recomendaciones relacionadas con XML.*

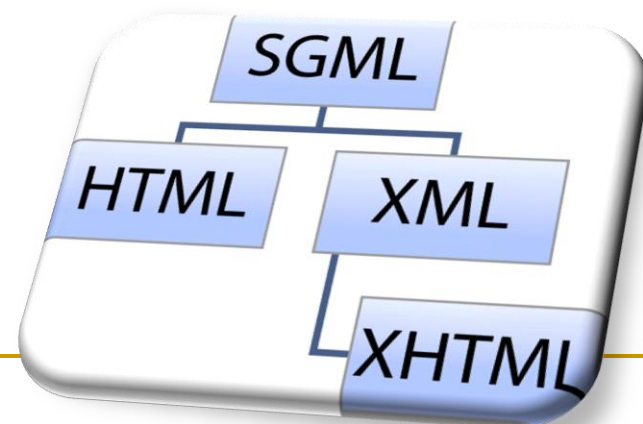


# Índice de Contenidos



# Breve historia de los lenguajes de marcas

- ❑ Desde que existen los ordenadores, los programadores han intentado crear modos de intercambio de información entre los diferentes programas informáticos. **SGML (Standard Generalized Markup Language)** fue uno de los primeros intentos en el que se pretendía combinar un **formato de dato universalmente intercambiable** con la posibilidad de **almacenar los datos con información acerca de su presentación y formato**. SGML fue creado para que se transformara en un estándar en el intercambio electrónico, manejo y publicación de documentos. Pero cuando se debe tratar con gran cantidad de datos complejos SGML se vuelve un lenguaje muy complicado.



# Breve historia de los lenguajes de marcas

*El documento sgml básico consiste en una DTD o Declaración de Tipo de Documento, una de las diversas "etiquetas" o marcas de nivel superior, párrafos y texto. La etiqueta de nivel superior puede ser <book>, <chapter>, <article>, o <sect1>. El siguiente es un ejemplo de un documento sgml sencillo:*

```
<!DOCTYPE article PUBLIC "-//OASIS//DTD DocBook V3.1//EN">  
<article>  
  <sect1 id="introduction">  
    <title>Introducción al hola mundo</title>  
    <para> ¡Hola mundo! </para>  
  </sect1>  
</article>
```

# Breve historia de los lenguajes de marcas

- ❑ **HTML es un ejemplo de un lenguaje definido en SGML.** HTML, tal y como fue concebido, era un lenguaje para el intercambio de documentos científicos y técnicos adaptado para su uso por no especialistas en tratamiento de documentos. **HTML resolvió el problema de la complejidad de SGML sirviéndose de un reducido conjunto de etiquetas estructurales y semánticas apropiadas para la realización de documentos relativamente simples.** Además de simplificar la estructura de los documentos, HTML **soportaba el hipertexto.** Las posibilidades de usar elementos multimedia fueron añadidas con posterioridad.
- ❑ XML fue concebido como un medio para recobrar la **potencia y flexibilidad** de SGML sin su gran complejidad. A pesar de ser una forma restringida de SGML, XML conserva casi toda la riqueza de las características de SGML. XML se considera un subconjunto de dicha especificación superior (**SGML o Standard Generalized Markup Language**), y que ha sido adaptada para el almacenamiento de datos.

# Consortio W3C

El organismo que establece los estándares es la **World Wide Web Consortium (W3C)**: Un organismo internacional de normalización, creado en 1994, similar en cierto sentido a ISO o ANSI. Sus miembros son más de 400 organizaciones de todo el mundo que corren con los gastos de financiación –aparte de algunos ingresos estatales- y lo componen profesionales de la informática de todos los sectores, cuya misión es la de definir y normalizar la utilización de los lenguajes usados en Internet, mediante un conjunto de Recommendations (recomendaciones) que son publicadas libremente, en su sitio Web [www.w3.org](http://www.w3.org) y aprobadas por comités de expertos compuestos por representantes nominales de la propia W3C y técnicos especializados de las más importantes compañías productoras de software para Internet, distribuidoras y centros de investigación.





# Consortio W3C

The screenshot shows the W3C website with a blue header containing the W3C logo and navigation links: STANDARDS, PARTICIPATE, MEMBERSHIP, and ABOUT W3C. A search bar with the Google logo is on the right. Below the header, there are three main sections: a left sidebar, a central news area, and a right sidebar.

**STANDARDS**

- Web Design and Applications
- Web Architecture
- Semantic Web
- XML Technology
- Web of Services
- Web of Devices
- Browsers and Authoring Tools
- ... or view all

**WEB FOR ALL**

- W3C A to Z
- Accessibility
- Internationalization

**News Articles:**

- W3C Workshop: Web Performance**  
28 September 2012 | [Archive](#)  
W3C announced today a *Workshop on Web Performance*, 8 November, hosted by Google at their Mountain View, California campus. As the Open Web Platform expands, the need for high performance implementation has grown, particularly on mobile devices. Participants will examine a broad range of performance issues and how they might be addressed. There is no fee to participate in this Workshop and W3C Membership is not required. All participants are required to submit a statement of interest by 29 October. Learn more about W3C's [Web Performance Working Group](#), which also invites people to share performance issues via their survey on [Open Web Platform Performance Priorities](#).
- W3C Workshop: Making the Multilingual Web work**  
02 October 2012 | [Archive](#)
- Two Drafts Published by the Tracking Protection Working Group**  
02 October 2012 | [Archive](#)
- Web API Design Cookbook Note Published**  
02 October 2012 | [Archive](#)

**Jobs:**

- [Software Quality Assurance \(QA\) Manager](#)
- [Packaged Web Applications Specialist](#)
- [Keio Business Development Lead](#)
- [Web Accessibility Specialist](#)
- [Web Accessibility Engineer](#)
- [Membership Manager at the W3C UK/Ireland Office](#)

**W3C BLOG**



# Consortio W3C

- ❑ La primera especificación relevante publicada por la W3C fue la versión *HTML 3.2*. La *última ha sido* muy reciente (26/Enero/2000) y corresponde a *XHTML 1.0*, que no es sino una reformulación de HTML 4.0, por lo que tiene, básicamente, las mismas funcionalidades, pero cumple las especificaciones más estrictas de XML.

*Incorrecto: <br> HTML 4.0*

*Correcto: <br></br> o <br/> o <br /> XHTML 1.0*

*Nota: Cualquiera de las tres formas es válida en XHTML*

```
<?xml version="1.0"
<!DOCTYPE html PUBLIC
<html xmlns="http://www.w3
  <head><title>XYZ</title>
  </head>
  <body>
  <p>
    voluptatem accusantium do
    totam rem aperiam eaque
  </p>
  </body>
</html>
```

**XHTML**

# Definición

- Así pues, se define el estándar XML como: *El formato universal para documentos y datos estructurados en Internet, y podemos explicar las características de su funcionamiento a través de 7 puntos importantes, tal y como la propia W3C recomienda:*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!doctype book PUBLIC "-//OASIS//DTD DocBook V4.1//EN">
<book lang=es>
  <bookinfo>
    <date>Mayo 2009</date>
    <title>Manual Usuario - Escuela Mundo Feliz</title>
    <subtitle>Registro de Alumnos</subtitle>
    <author>
      <firstname>Henry</firstname>
      <surname>Molina Noboa</surname>
    </author>
  </bookinfo>
  <chapter>
    <title>Introducción</title>
    <para>El registro de alumnos permite dar de alta a los alumnos en la Escuela Mundo Feliz.</para>
  </chapter>
  <chapter>
    <title>Objetivo</title>
    <para>Explicar el uso de la pantalla de registro de estudiantes.</para>
  </chapter>
</book>
```

# Definición

***XML es un estándar para escribir datos estructurados en un fichero de texto.***

***XML parece HTML pero no lo es.***

***XML está en formato texto, pero no para ser leído.***

***XML consta de una familia de tecnologías.***

***XML es prolijo, pero eso no supone un problema.***

***XML es nuevo, pero no tanto.***

***XML no requiere licencias, es independiente de la plataforma, y tiene un amplio soporte.***

# Definición

***XML es un estándar para escribir datos estructurados en un fichero de texto.***

En la práctica, la mayor parte de información útil no aparece aislada en forma de datos simples, sino que lo hace de forma organizada y estructurada. Los diccionarios, guías, enciclopedias, etc., son colecciones de datos que serían inútiles si no estuvieran organizadas de acuerdo con unas determinadas reglas. Además, tener estructurada la información supone ventajas adicionales, al facilitar el acceso y el manejo de los datos. Por ello parece razonable desarrollar la idea de la agrupación de datos, que tengan un cierto tipo de estructura y organización interna. XML es un conjunto de reglas, normas y convenciones para diseñar formatos de texto para tales **tipos de datos**, de forma que produzca ficheros fáciles de generar y de leer.

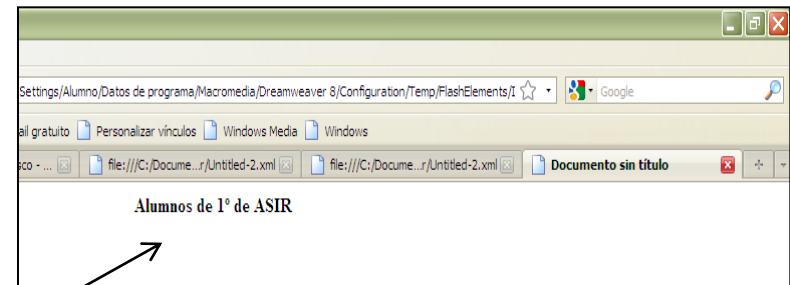


# Definición

**XML parece HTML pero no lo es.**

XML usa **marcas y atributos**, pero la diferencia estriba en que, mientras en HTML cada marca y atributo está establecido mediante un significado, incluyendo el aspecto que debe tener al verse en un navegador, en XML sólo se usan las marcas para delimitar fragmentos de datos, dejando la interpretación de éstos a la aplicación que los lee.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html >  
<head>  
<meta http-equiv="Content-Type" content="text/html;  
charset=iso-8859-1" />  
<title>Documento sin título</title>  
</head>  
<body>  
<div align="center"><strong>Alumnos de 1º de ASIR</strong></div>  
</body>  
</html>
```



**<strong>..</strong>** tiene como significado: visualizar el fragmento de datos en negrita

# Definición

**XML parece HTML pero no lo es.**

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<curso>
```

```
<alumno>
```

```
<nombre>Juan Antonio</nombre>
```

```
<apellidos>López Pérez</apellidos>
```

```
<dni>23252300</dni>
```

```
</alumno>
```

```
<alumno>
```

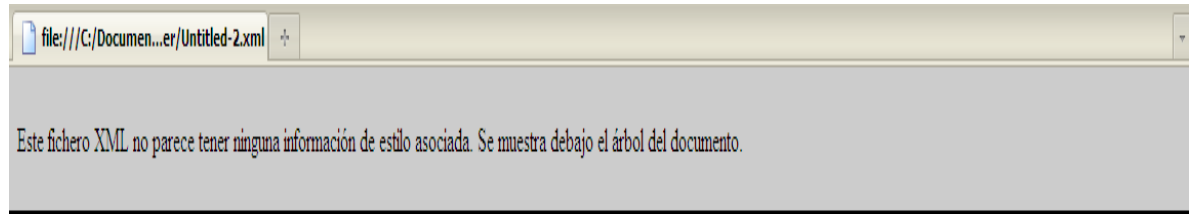
```
<nombre>José Antonio</nombre>
```

```
<apellidos>Sánchez Pérez</apellidos>
```

```
<dni>23285900</dni>
```

```
</alumno>
```

```
</curso>
```



```
-<curso>  
  -<alumno>  
    <nombre>Juan Antonio</nombre>  
    <apellidos>López Pérez</apellidos>  
    <dni>23252300</dni>  
  </alumno>  
  -<alumno>  
    <nombre>José Antonio</nombre>  
    <apellidos>Sánchez Pérez</apellidos>  
    <dni>23285900</dni>  
  </alumno>  
</curso>
```

Los documentos XML deben tener una estructura jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Esto significa que los elementos deben estar correctamente anidados y que los elementos no se pueden solapar entre ellos. Además los elementos con contenido deben estar correctamente cerrados.

# Definición

**XML está en formato texto, pero no para ser leído.**

- ❑ Esto le da innumerables ventajas de portabilidad, depuración, independencia de plataforma, e incluso de edición, pero su sintaxis es **más estricta que la de HTML**: una marca olvidada o un valor de atributo sin comillas convierten el documento en inutilizable. *No hay permisividad en la construcción de documentos, ya que esa es la única forma de protegerse contra problemas más graves.*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml">  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=iso-  
8859-1" />  
<title>Documento sin t&iacute;tulo</title>  
</head>  
<body>  
<div align="center">  
<p><strong>Alumnos de 1&ordm; de ASIR</strong></p>  
<p>Hemos eliminado una etiqueta </p>  
</body>  
</html>
```

Alumnos de 1º de ASIR

Hemos eliminado una etiqueta

Falta </div>



# Definición

***XML está en formato texto, pero no para ser leído.***

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<curso>
```

```
<alumno>
```

```
<nombre>Juan Antonio</nombre>
```

```
<apellidos>López Pérez</apellidos>
```

```
<dni>23252300</dni>
```

```
</alumno>
```

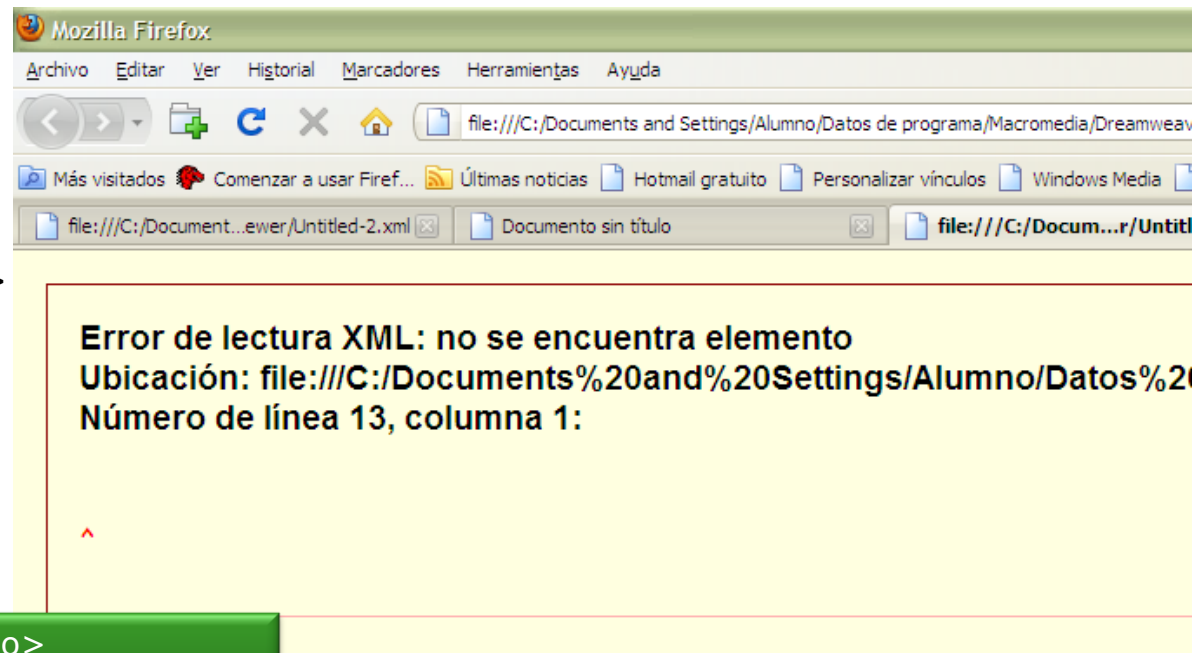
```
<alumno>
```

```
<nombre>José Antonio</nombre>
```

```
<apellidos>Sánchez Pérez</apellidos>
```

```
<dni>23285900</dni>
```

```
</alumno>
```



# Definición

***XML consta de una familia de tecnologías.***

Por supuesto, existe una definición (estándar) de **XML 1.0** que **viene de Febrero 98**, pero su desarrollo se ha ido enriqueciendo paulatinamente a medida que se veían sus posibilidades, como ejemplo tenemos **XSL** que es precisamente, una extensión del lenguaje de hojas de estilo (CSS) , y *basado en la combinación de tres tecnologías: XSLT - a language for transforming XML documents XPath - a language for navigating in XML documents XSL-FO.* Y otros muchas tecnologías....



# Definición

**XML consta de una familia de tecnologías.**



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="estilo_catalogo.css"?>
<catalog>
  <cd>
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <country>USA</country>
    <company>Columbia</company>
    <price>10.90</price>
    <year>1985</year>
  </cd>
  <cd>
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <country>UK</country>
    <company>CBS Records</company>
    <price>9.90</price>
    <year>1988</year>
  </cd>
</catalog>
```

catalogo.xml

```
cd
{
  display:block;
  width: 350 px;
  padding: 10px;
  margin-bottom:10px;
  border: 4px double black;
  background-color: silver;
  color: blue;
  text-align: center;
}
title
{
  display:inline;
  font-family: Times, serif;
  font-size: 16pt;
}
artist, country, company,
price, year {display:none;}
```

estilo\_catalogo.css

# Definición

**XML consta de una familia de tecnologías.**



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="estilo_catalogo.xsl"?>
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist >Bonnie Tyler</artist>
<country>UK</country>
<company>CBS Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
</catalog>
```

catalogo.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<link rel="stylesheet" href="miestilo.css"/>
</head>
<body>
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<tr>
<td><xsl:value-of select="catalog/cd/title"/></td>
<td><xsl:value-of select="catalog/cd/artist"/></td>
</tr>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

estilo\_catalogo.xsl

# Definición

**XML consta de una familia de tecnologías.**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="estilo_catagolo_2.xsl"?>
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist >Bonnie Tyler</artist>
<country>UK</country>
<company>CBS Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
</catalog>
```

catalogo.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <html>
  <body>
  <h2>My CD Collection</h2>
  <table border="1">
    <tr bgcolor="#9acd32">
      <th>Title</th>
      <th>Artist</th>
    </tr>
    <xsl:for-each select="catalog/cd">
      <tr>
        <td><xsl:value-of select="title"/></td>
        <td><xsl:value-of select="artist"/></td>
      </tr>
    </xsl:for-each>
  </table>
  </body>
  </html>
</xsl:template>
</xsl:stylesheet>
```

estilo\_catagolo\_2.xsl

# Definición

**XML consta de una familia de tecnologías.**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl"
href="estilo_catagolo_3.xsl"?>
<catalog>
<cd>
<title>Empire Burlesque</title>
<artist>Bob Dylan</artist>
<country>USA</country>
<company>Columbia</company>
<price>10.90</price>
<year>1985</year>
</cd>
<cd>
<title>Hide your heart</title>
<artist >Bonnie Tyler</artist>
<country>UK</country>
<company>CBS Records</company>
<price>9.90</price>
<year>1988</year>
</cd>
</catalog>
```

catalogo.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<head>
<link rel="stylesheet" href="miestilo.css"/>
</head>
<body>
<div class="cuerpo">
<h2>My CD Collection</h2>
<table border="1">
<tr bgcolor="#9acd32">
<th>Title</th>
<th>Artist</th>
</tr>
<xsl:for-each select="catalog/cd">
<tr>
<td><xsl:value-of select="title"/></td>
<td><xsl:value-of select="artist"/></td>
</tr>
</xsl:for-each>
</table>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

estilo\_catagolo\_3.xsl

```
div.cuerpo
{
display:block;
width: 350 px;
padding: 10px;
margin-bottom:10px;
border: 4px double black;
background-color: silver;
color: blue;
text-align: center;
}
h2
{
color: purple;
}
td
{
font-family: Times, serif;
font-size: 16pt;
}
```

miestilo.css

# Definición

***XML consta de una familia de tecnologías.***

El Modelo de Objetos de Documento (**DOM**) es un conjunto estándar de funciones para manipular documentos XML (y HTML) mediante un lenguaje de programación (***alert(document.title);***).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Documento sin título</title>
<script language="javascript">
alert(document.title);
</script>
</head>
<body>
<div align="center">
<p><strong>Alumnos de 1º de ASIR</strong></p>
<p>Hemos eliminado una etiqueta </p>
</body>
</html>
```



# Definición

**XML consta de una familia de tecnologías.**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
```

Books.xml

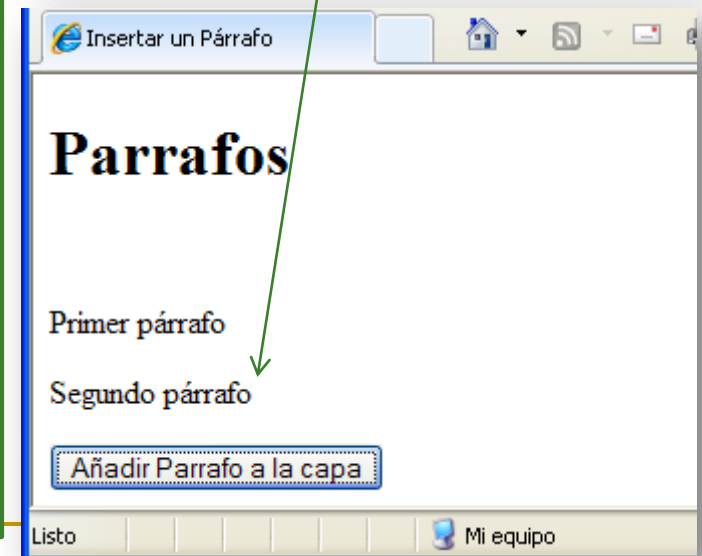
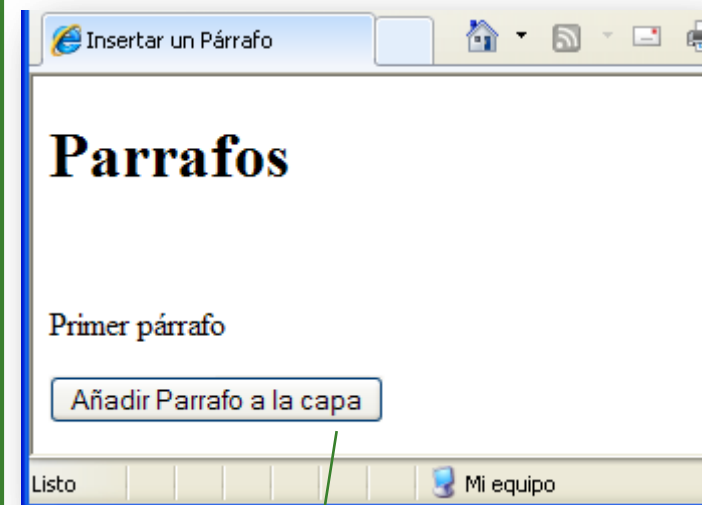
```
<book category="web">
  <title lang="en">XQuery Kick Start</title>
  <author>James McGovern</author>
  <author>Per Bothner</author>
  <author>Kurt Cagle</author>
  <author>James Linn</author>
  <author>Vaidyanathan Nagarajan</author>
  <year>2003</year>
  <price>49.99</price>
</book>
<book category="web" cover="paperback">
  <title lang="en">Learning XML</title>
  <author>Erik T. Ray</author>
  <year>2003</year>
  <price>39.95</price>
</book>
</bookstore>
```

<http://www.w3schools.com/dom/default.asp>

# Definición

*XML consta de una familia de tecnologías.*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Insertar un Párrafo</title>
<script language="javascript">
function insertar_p()
{
    var con=document.getElementById("contenido");
    var p=document.createElement("p");
    var texto=document.createTextNode("Segundo párrafo");
    p.appendChild(texto);
    //document.body.insertBefore(p,con);
    con.appendChild(p);
}
</script>
</head>
<body>
<h1 id="inicio">Párrafos</h1>
<div id="contenido">
<p>Primer párrafo</p>
</div>
<form action="" method="get">
<input name="" type="button" value="Añadir Párrafo a la capa"
onClick="insertar_p()" />
</form>
</body>
</html>
```



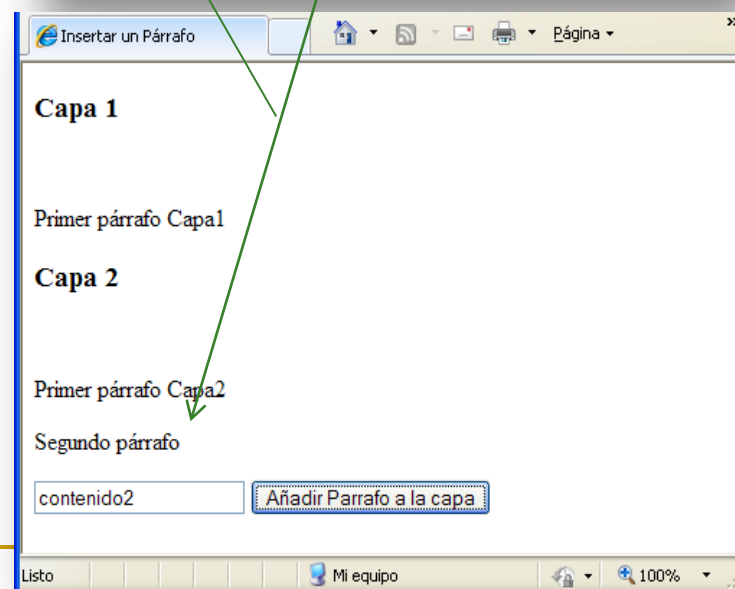
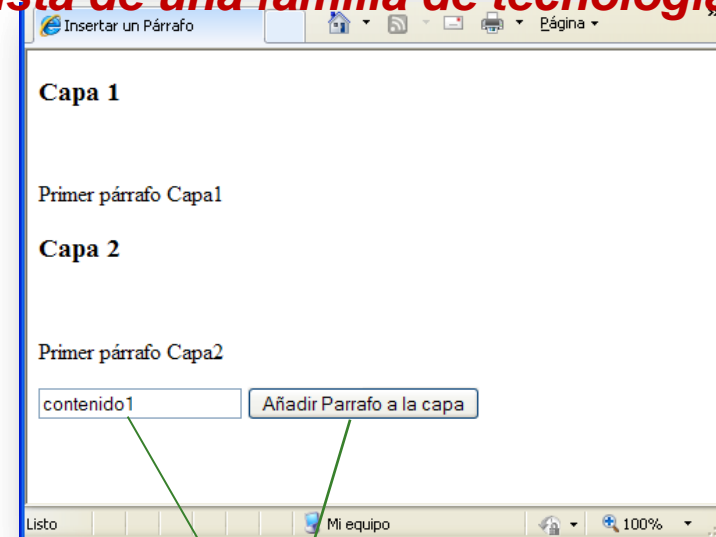
```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Insertar un Párrafo</title>
<script language="javascript">
function insertar_p(capa)
{
    var con=document.getElementById(capa);
    var p=document.createElement("p");
    var texto=document.createTextNode("Segundo párrafo");
    p.appendChild(texto);
    con.appendChild(p);
}
</script>
</head>
<body>
<div id="contenido1">
<h3 id="inicio">Capa 1</h3>
<p>Primer párrafo Capa1</p>
</div>
<div id="contenido2">
<h3 id="inicio">Capa 2</h3>
<p>Primer párrafo Capa2</p>
</div>
<form action="" method="get">
<input id="capa" type="text" value="contenido1" />
<input type="button" value="Añadir Párrafo a la capa"
onClick="insertar_p(document.getElementById('capa').value)" />
</form>
</body>
</html>

```

# Definición

**XML consta de una familia de tecnologías.**



# Definición

***XML es prolijo, pero eso no supone un problema.***

Los ficheros resultantes, son casi siempre mayores que sus equivalentes binarios. Dado el reducido coste actual del espacio en disco y la existencia gratuita de utilidades de compresión, junto al hecho de que los protocolos de comunicación soportan sistemas rápidos de compresión, este aspecto no debe resultar problemático.

***XML es nuevo, pero no tanto.***

El estándar empezó a diseñarse en 1996, y se publicó la recomendación en Febrero/98. Como ya hemos comentado, eso no significa que la tecnología no esté suficientemente madura, ya que el estándar SGML en el que se basa, data de una especificación ISO del año 1986.

# Definición

***XML no requiere licencias, es independiente de la plataforma, y tiene un amplio soporte.***

La selección de XML como soporte de aplicaciones, significa entrar en una comunidad muy amplia de herramientas y desarrolladores, y en cierto modo, se parece a la elección de SQL respecto a las bases de datos. Todavía hay que utilizar herramientas de desarrollo, pero la tranquilidad del uso del estándar y de su formato, hacen que las ventajas a la larga sean notables.

- Tenemos pues, dos partes bien definidas dentro de todo documento XML: la *definición de contenidos* y los propios contenidos (el DTD/XSD y los datos). Cada definición, o DTD/XSD constituye de por sí una forma de escribir documentos para Internet.

# Definición

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE lista_de_personas [  
<!ELEMENT lista_de_personas (persona*)>  
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>  
<!ELEMENT nombre (#PCDATA) >  
<!ELEMENT fechanacimiento (#PCDATA) >  
<!ELEMENT sexo (#PCDATA) >  
<!ELEMENT numeroseguridadsocial (#PCDATA)>  
>  
>
```



definición de contenidos

```
<lista_de_personas>  
  <persona>  
    <nombre>José García</nombre>  
    <fechanacimiento>25/04/1984</fechanacimiento>  
    <sexo>Varón</sexo>  
  </persona>  
</lista_de_personas>
```

<http://www.xmlvalidation.com/>

# Definición

www.xmlvalidation.com/documentation.html

Google Gmail: correo electr... Universidad de Mur... Sakai


```
13 <el...
14 <link manager="Big.Boss"/>
15 </person>
```

## An example

If you would like to see how xmlvalidation.com works, but don't have a testing, you can use [this example](#).

First, paste or upload the XML file. You will then be prompted for the XSD uploaded, the errors will be highlighted and described.

Please copy your XML document in here:



Please copy your XML document in here:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas [
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA) >
]>
```

Or upload it:

No se ha seleccionado ningún archivo

The validation check is performed against any XML schema or DTD declared inside the XML document.

If neither an XML schema nor a DTD is declared, only a syntax check is performed.

To validate the XML document against an external XML schema, click below.

Validate against external XML schema

Or upload it:

No se ha seleccionado ningún archivo

The validation check is performed against any XML schema or DTD declared inside the XML document.

If neither an XML schema nor a DTD is declared, only a syntax check is performed.

To validate the XML document against an external XML schema,

Validate against external XML schema

XML

You can read more about XML on <http://www.w3.org/XML/>.

## Validate an XML file

Read here how to [validate your XML files \(including referenced DTDs\)](#) online with just a few mouse clicks.

**No errors were found**



# Definición

the "Validate" button:

Filename:

**Note:** If you get an "Access denied" error, it's because your browser security does not allow file access across domains.

The file "note\_error.xml" demonstrates your browsers error handling. If you want to see an error free message, substitute the "note\_error.xml" with "cd\_catalog.xml".

---

## Validate Your XML Against a DTD

If you run **Internet Explorer**, you can validate your XML against a DTD in the text area below.

Just add the DOCTYPE declaration (with the DTD) after the <xml> element, and click the "Validate" button:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas [
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?,
numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
]>
<lista_de_personas>
<persona>
  <fechanacimiento>25/04/1984</fechanacimiento>
  <sexo>Varón</sexo>
</persona>
</lista_de_personas>
```

# Definición

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE lista_de_personas [
```

```
<!ELEMENT lista_de_personas (persona*)>
```

```
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
```

```
<!ELEMENT nombre (#PCDATA) >
```

```
<!ELEMENT fechanacimiento (#PCDATA) >
```

```
<!ELEMENT sexo (#PCDATA) >
```

```
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

```
]>
```

```
<lista_de_personas>
```

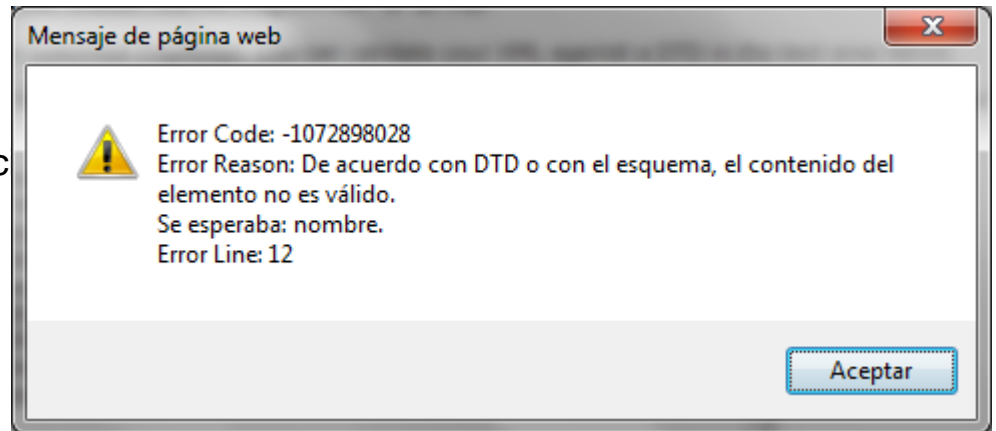
```
<persona>
```

```
<fechanacimiento>25/04/1984</fechanac
```

```
<sexo>Varón</sexo>
```

```
</persona>
```

```
</lista_de_personas>
```



# Definición

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas SYSTEM "docValidarSchema.dtd">
<lista_de_personas>
  <persona>
    <nombre>José García</nombre>
    <fechanacimiento>25/04/1984</fechanacimiento>
    <sexo>Varón</sexo>
  </persona>
</lista_de_personas>
```



definición de contenidos

## Fichero docValidarSchema.dtd

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

# Definición: *Mi primer (X)HTML*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
```

```
<title>Documento sin título</title>
```

```
</head>
```

```
<body>
```

```
</body>
```

```
</html>
```



# Definición: *Mi primer (X)HTML*

← → ↻ [www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd](http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd)

You can syntax-check an XML file by typing the URL of the file into the input field below, and then click the "Validate" button:

Validate (IE Only!)

[« Previous](#)

[Next Chapter »](#)

xhtml1-transitional (1).dtd

Mostrar tod

# Definición: *Mi primer (X)HTML*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
```

Todos los documentos XHTML válidos deben llevar un elemento llamado DOCTYPE, el cual no es parte del documento en sí, sino que define el tipo de DTD (Document Type Definition o Definición de tipo de documento) a emplear en nuestros documentos, es obligatorio y puede ser uno de estos tres:

- XHTML 1.0 Strict: Se usa cuando se desea utilizar al 100% XHTML, su nombre lo dice bien claro, es XHTML estricto, la declaración del mismo es como sigue:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- **HTML 1.0 Transitional: Es el más usado ya que permite manejar elementos de XHTML y HTML 4.01, su declaración es la que sigue:**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

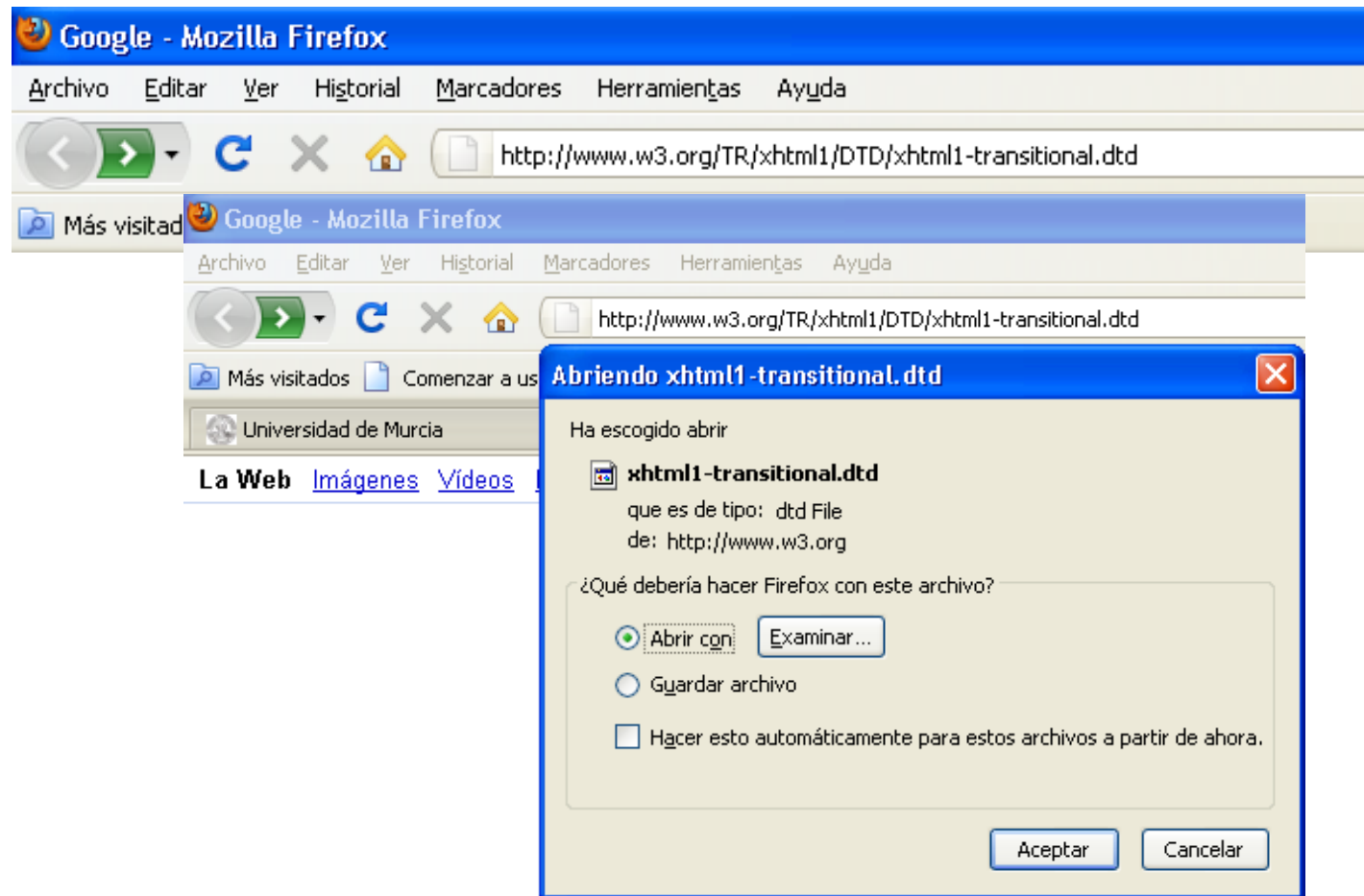
- XHTML 1.0 Frameset: Se debe usar cuando se manejan frames, su declaración es la siguiente:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

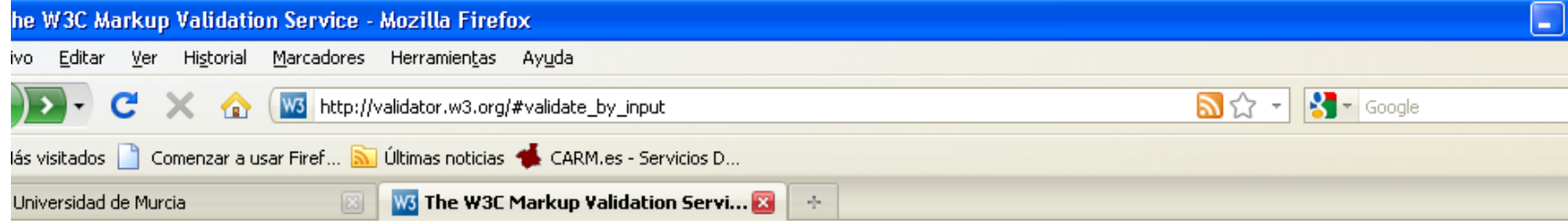
Los frames (marcos o cuadros) permiten dividir la ventana en varias más pequeñas, de modo que en cada una de ellas se cargue una página html distinta.

# Definición: *Mi primer (X)HTML*

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd>
```



# Definición: *Mi primer (X)HTML*



Validate by URI    Validate by File Upload    **Validate by Direct Input**

### Validate by direct input

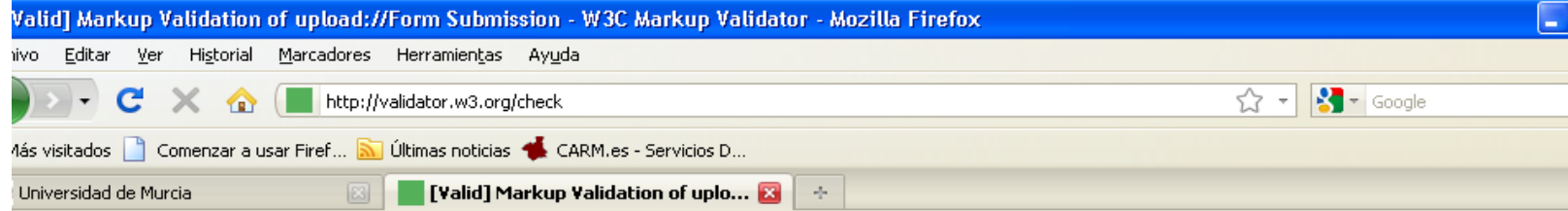
Enter the Markup to validate:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Documento sin título</title>
</head>
<body>
</body>
</html>
```

[More Options](#)



# Definición: *Mi primer (X)HTML*



Jump To: [Notes and Potential Issues](#) [Congratulations · Icons](#)

This document was successfully checked as XHTML 1.0 Transitional!

<b>Result:</b>	Passed, <b>2 warning(s)</b>	
<b>Source:</b>	<pre>&lt;!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"&gt; &lt;html xmlns="http://www.w3.org/1999/xhtml"&gt; &lt;head&gt; &lt;title&gt;Documento sin título&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;/body&gt; &lt;/html&gt;</pre>	
<b>Encoding:</b>	utf-8	(detect automatically) ▾
<b>Doctype:</b>	XHTML 1.0 Transitional	(detect automatically) ▾
<b>Root Element:</b>	html	
<b>Root Namespace:</b>	<a href="http://www.w3.org/1999/xhtml">http://www.w3.org/1999/xhtml</a>	



# Definición: *Mi primer (X)HTML*

## Congratulations

The uploaded document was successfully checked as XHTML 1.0 Transitional. This means that the resource in question identified itself as "XHTML 1.0 Transitional" and that we successfully performed a formal validation using an SGML, HTML5 and/or XML Parser(s) (depending on the markup language used).

### *"valid" Icon(s) on your Web page*

To show your readers that you have taken the care to create an interoperable Web page, you may display this icon on any page that validates. Here is the HTML you could use to add this icon to your Web page:



```
<p>  
<a href="http://validator.w3.org/check?uri=referer"></a>  
</p>
```

A [full list](#) of icons, with links to alternate formats and colors, is available: If you like, you can download a copy of the icons to keep in your local web directory, and change the HTML fragment above to reference your local image rather than the one on this server.

### *Validating CSS Style Sheets*

If you use [CSS](#) in your document, you can check it using the W3C [CSS Validation Service](#).

```
<p>  
<a href="http://validator.w3.org/check?uri=referer">  
  
</a>  
</p>
```

# Definición: *Mi primer (X)HTML*

The image shows two browser windows. The top window is Windows Internet Explorer, displaying a local file named 'Documento sin título' at the path 'C:\Documents and Settings\Profesor\Escritorio\Untitled-1.html'. The bottom window is Mozilla Firefox, displaying the SUMA 2.0 website at 'https://suma.um.es/suma/sumav2/'. The website features a login form with fields for 'Modo de acceso' (set to 'Correo'), 'Usuario' (with placeholder 'Indique su e-mail'), and 'Contraseña', and an 'Acceder' button. A navigation menu includes links for '¿Qué es SUMA?', 'Modos de Acceso', 'Oferta de Cursos', 'Guías de SUMA', '¿Problemas?', and 'Apoyo a la Teleenseñanza'. The footer contains 'RECOMENDACIONES DE NAVEGACIÓN | CONTACTAR | INCIDENCIAS | SUGERENCIAS' and validation logos for W3C XHTML 1.0, W3C CSS, and W3C WAI-AA WCAG 1.0.

Documento sin título - Windows Internet Explorer

C:\Documents and Settings\Profesor\Escritorio\Untitled-1.html

Google

Archivo Edición Ver Favoritos Herramientas Ayuda

Google

Buscar

Compartir

Acceder

Favoritos

Universidad de Murcia (17) Universidad de Murcia (16) Universidad de Murcia (15) Universidad de Murcia (14)

Documento sin título

W3C XHTML 1.0

SUMA: Campus virtual de la Universidad de Murcia - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

https://suma.um.es/suma/sumav2/

Más visitados Comenzar a usar Firef... Últimas noticias CARM.es - Servicios D...

Universidad de Murcia [Valid] Markup Validation of upload://Fo... SUMA: Campus virtual de la Univ...

suma2.0 CAMPUS VIRTUAL

UNIVERSIDAD DE MURCIA

**AVISO:** Desde el 09/09/2009 ha cambiado el correo de los alumnos usuario@alu.um.es por usuario@um.es. A partir de esta fecha utilice **usuario@um.es** para conectarse a su cuenta de correo o a las aplicaciones de la Universidad de Murcia. Mas información en <http://www.um.es/etica/faq-unificacion-emails-umu>

Modo de acceso Correo

Usuario Indique su e-mail

Contraseña

Acceder

Bienvenidos a SUMA 2.0

- ¿Qué es SUMA?
- Modos de Acceso
- Oferta de Cursos
- Guías de SUMA
- ¿Problemas?
- Apoyo a la Teleenseñanza

RECOMENDACIONES DE NAVEGACIÓN | CONTACTAR | INCIDENCIAS | SUGERENCIAS

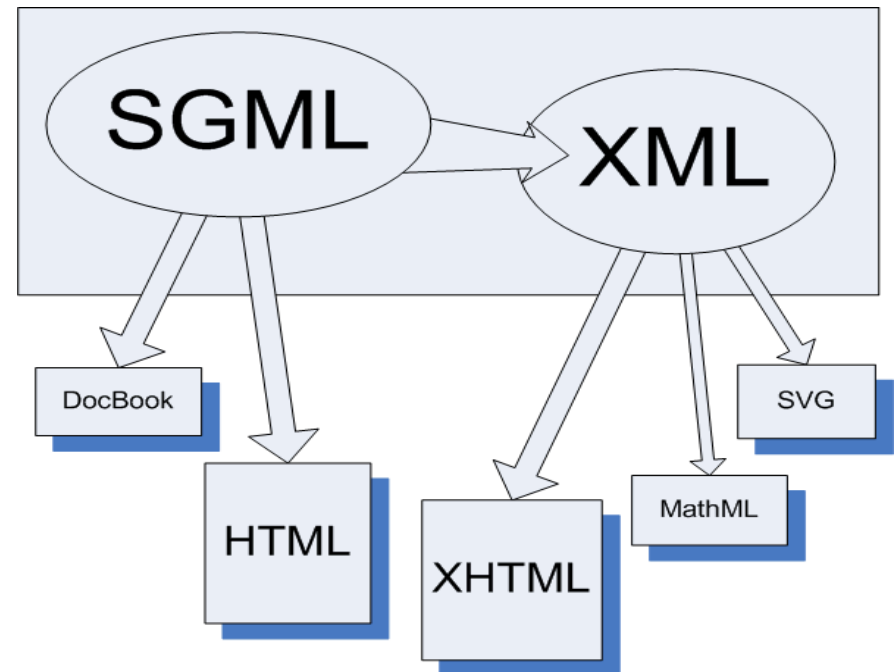
W3C XHTML 1.0 W3C CSS W3C WAI-AA WCAG 1.0

<https://suma.um.es/suma/sumav2/>

# HTML y XML

Por supuesto existe una diferencia fundamental entre XML y HTML:

- ◆ HTML está diseñado para una aplicación específica; para **transmitir la información hacia los seres humanos, usualmente, de manera visual** y por medio de un navegador Web.
- ◆ XML **no tiene una aplicación específica**; está diseñado para el uso que se precise. HTML tiene un conjunto finito de estructuras de marcas (<p>, <ul>, <h2>, etc.), que se utilizan para crear un documento HTML correcto.



# HTM y XML

- En teoría, podemos estar seguros que cualquier navegador nos mostrará un documento HTML correcto porque lo único que tiene que hacer es conocer este conjunto limitado de etiquetas. Aunque en la práctica nos encontramos con documentos WEB que son bien visualizados correctamente en un navegador e incorrectamente en otros, porque difieren en la utilización de algunas de las marcas que se utilizan en HTML.
- Por otra parte, si creamos un documento XML podemos estar seguros que cualquier **analizador (parser)** podrá extraer información del documento, pero lo que no podemos garantizar, es que todas las aplicaciones *comprendan el significado de esa información.*

# DTD (Document Type Definitions) / XML Schema (xsd)

- ❑ Si queremos tener la seguridad de que un fichero XML pueda **ser perfectamente interpretado por cualquier herramienta**, podemos (aunque no es obligatorio) incluir una definición de su construcción que preceda a los datos propiamente dichos. Este conjunto de meta-datos recibe el nombre de DTD (*Definición de Tipo de Documento*) o XML Schema (*xsd*).
- ❑ Así, si una herramienta es capaz de interpretar XML, eso significa que posee un analizador sintáctico (*Parser*) que es capaz de contrastar la definición dada por el autor del documento contra la especificada por la normativa (dtd/xsd), indicando si hay errores, y, de no ser así, presentando/procesando de forma adecuada.

# Las DTD (Document Type Definitions)

Macromedia Dreamweaver 8 - [F:\IES 2010-2011\LMSGI\WT 3. Introducción al Lenguaje de Marcas - XML\Ejemplo\ejemplo\_DTD.xml\*]

Archivo Edición Ver Insertar Modificar Texto Comandos Sitio Ventana Ayuda

Común

ejemplo\_DTD.xml\*

Código Dividir Diseño Título:

```
1 <?xml version="1.0" encoding="ISO-8859-1"?>
2
3 <!DOCTYPE lista_de_personas [
4 <!ELEMENT lista_de_personas (persona*)>
5 <!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
6 <!ELEMENT nombre (#PCDATA) >
7 <!ELEMENT fechanacimiento (#PCDATA) >
8 <!ELEMENT sexo (#PCDATA) >
9 <!ELEMENT numeroseguridadsocial (#PCDATA)>
10 ]>
11
12 <lista_de_personas>
13   <persona>
14     <nombre>José García</nombre>
15     <fechanacimiento>25/04/1984</fechanacimiento>
16     <sexo>Varón</sexo>
17   </persona>
18 </lista_de_personas>
19
```

<lista de personas> <persona> <sexo>

1K / 1 s

Propiedades

Resultados

Archivo	Lí...	Descripción
F:\IES 2010-2011\LMSGI...	16	Unknown element 'sexo'[xml]
F:\IES 2010-2011\LMSGI...	17	Element 'sexo' is not valid for content model: '(nombre,fechanacimiento?,sexo?,numeroseguridadsocial?)[xml]

# Un primer ejemplo: "Hola mundo" en XML

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE Mensaje [ <!ELEMENT Contenido (#PCDATA)> ]>  
<!-- este es un comentario -->  
<Contenido>¡Hola, mundo!</Contenido>
```

- En el ejemplo podemos observar 3 líneas clave: La primera, es la definición general. Nos indica que lo que viene a continuación es un documento XML (las de inicio y fin son el carácter obligatorio que delimita esa definición). Además, observamos dos atributos: *versión* - que se establece a 1.0- que nos indica que el intérprete de XML debe de utilizar las normas establecidas en Febrero/98 y *encoding*, asignado a "UTF-8", y que el estándar recomienda incluir siempre, aunque algunos navegadores no lo exijan de forma explícita.





# Un primer ejemplo: "Hola mundo" en XML

- La segunda línea es una DTD muy simple. Consta de la declaración de 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 texto (#PCDATA).



# Un primer ejemplo: "Hola mundo" en XML

- ❑ Finalmente, la cuarta línea (la tercera es un simple comentario) contiene la información en sí. Dentro de dos etiquetas de apertura y cierre con el nombre definido en la línea 2, se incluye la información propiamente dicha..
- ❑ Hay que indicar que no estamos indicando ningún modo de presentación. Por tanto IE asume que lo que queremos es analizar el documento con el *parser* y *averiguar si existe algún error en él*: Reconoce el tipo de documento, simplifica el DTD limitándose a mostrar su cabecera, y recorre los datos cambiando el color de las marcas y símbolos para que la interpretación sea más sencilla.



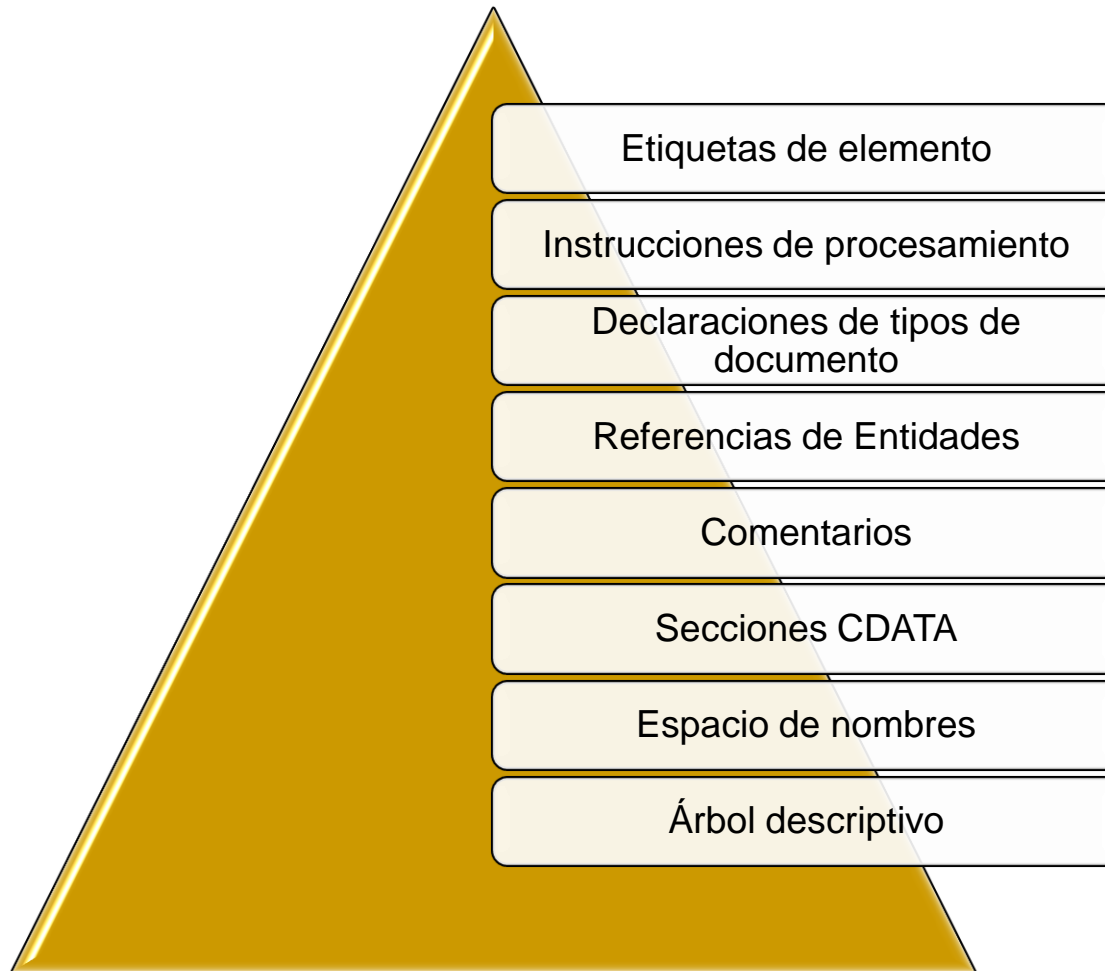
# Una vista al XML: Introducción

- ❑ Ya se ha adelantó la definición de documento XML como una información jerarquizada, en forma de texto, que constituye un objeto de datos que puede ser presentado mediante una estructura de árbol, que puede estar almacenado en un único archivo o estar dividido en varios.
- ❑ Para crear un documento XML se puede utilizar cualquier editor especializado; son cada vez más los paquetes de software que facilitan que sus datos puedan salvarse como tales documentos XML.
- ❑ La presentación textual de un documento XML, como en todo ML, se puede resumir de la forma : Texto XML = datos + marcado. Esto significa que el texto de un documento XML consta de dos conjuntos : marcado y datos. El marcado corresponde a las instrucciones que el analizador XML debe procesar ( que se incluyen entre los paréntesis angulares ) mientras que los datos son el texto entre las marca o etiqueta delimitada, en inicio y final por paréntesis angulares. El procesador, una vez determinado que todos los caracteres de un documento son aceptables, los diferencia entre texto de marcado y caracteres de datos ( CDATA ).

# Una vista al XML: Introducción

- ❑ Es importante reseñar que desde el principio debe distinguirse entre datos analizables ( Parsed Character Data o PCDATA ) y no analizables, y que su mezcla, en principio, no es un problema ya que admite esta posibilidad sin problema alguno, de la misma forma que en un texto en castellano se puede incluir un texto inglés, siempre que se señale adecuadamente.
- ❑ Los caracteres de datos corresponden a todo lo que no es marcado. La secuencia es: un inicio `< >`, seguido de un contenido, y de una finalización `< / >` ( la única excepción son las referencias a entidades que, como veremos, comienzan con el carácter “&”, y terminan con el carácter “;”).

# Una vista al XML: Sintaxis



# Una vista al XML: Sintaxis

## Etiquetas de Elemento

- ❑ Las etiquetas constituye el componente más evidente de la sintaxis XML y se emplean para describir elementos. `<ciudad>` y `</ciudad>`. y se estructuran de manera jerárquica con un único elemento raíz.

*Los elementos XML pueden tener contenido (más elementos, caracteres, o ambos a la vez), o bien ser elementos vacíos.*

*Un elemento con contenido es, por ejemplo:*

*`<nombre>Fernando Damián</nombre>`*

*`<aviso tipo="emergencia" gravedad="mortal">Que no cunda el pánico</aviso>`*

*Siempre empieza con una `<etiqueta>` que puede contener atributos o no, y termina con una `</etiqueta>` que debe tener el mismo nombre.*

# Una vista al XML: Sintaxis

## Etiquetas de Elemento

Un elemento puede estar vacío, por lo que no tiene contenido. Por ejemplo;

```
<identificador DNI="23123244"/>
```

```
<linea-horizontal/>
```

Al no tener una etiqueta de cierre que delimite un contenido, se utiliza la forma `<etiqueta/>`, que puede contener atributos o no.

Los elementos pueden tener atributos, que son una manera de incorporar características o propiedades a los elementos de un documento.

Por ejemplo, un elemento "chiste" puede tener un atributo "tipo" y un atributo "calidad", con valores "vascos" y "bueno" respectivamente.

```
<chiste tipo="vascos" calidad="bueno"> Esto es un día que Patxi y Josu van  
paseando... </chiste>
```

# Una vista al XML: Sintaxis

## Etiquetas de Elemento

Al igual que en otras cadenas literales de XML, los atributos pueden estar marcados entre comillas simples (') o doble ("). Cuando se usa uno para delimitar el valor del atributo, el otro tipo se puede usar dentro.

```
<verdura clase="zanahoria" logitud='15" y media'>
```

```
<cita texto=""'Hola buenos dias', dijo él">
```

A veces, un elemento con contenido, puede modelarse como un elemento vacío con atributos.

```
<gato><nombre>Micifú</nombre><raza>Persa</raza></gato>
```

```
<gato raza="Persa">Micifú</gato>
```

```
<gato raza="Persa" nombre="Micifú"/>
```



# Una vista al XML: Sintaxis

## Instrucciones de procesamiento

```
<?xml version=" 1.0 " encoding=" UTF-8 " standalone=" yes "?>  
<ficha>  
<nombre> Angel </nombre>  
<apellido> Barbero </apellido>  
<direccion> c/Ulises, 36 </direccion>  
</ficha>
```

Lo primero que tenemos que observar es la primera línea. Con ella deben empezar todos los documentos XML, ya que es la que indica que lo que la sigue es XML. Aunque es opcional, es recomendable incluirla. Puede tener varios atributos, algunos obligatorios y otros no:

**version:** Indica la versión de XML usada en el documento. Es obligatorio ponerlo, a no ser que sea un documento externo a otro que ya lo incluía.

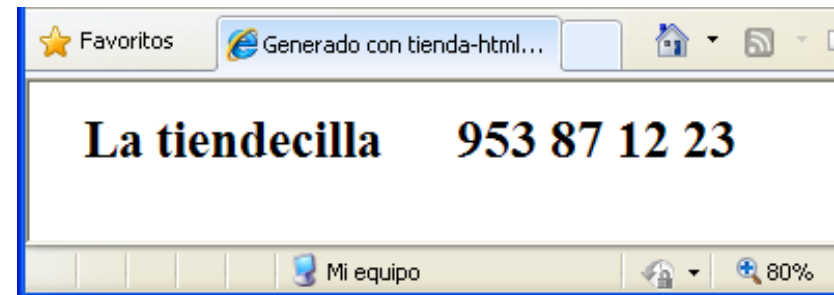
**encoding:** La forma en que se ha codificado el documento. Se puede poner cualquiera, y depende del parser el entender o no la codificación. Por defecto es UTF-8, aunque podrían ponerse otras, como UTF-16, US-ASCII, ISO-8859-1, etc.

**standalone:** Indica si el documento va acompañado de un DTD ("no"), o no lo necesita ("yes"); en principio no hay porqué ponerlo, porque luego se indica el DTD si se necesita.

# Una vista al XML: Sintaxis

## Instrucciones de procesamiento

```
<?xml version="1.0" encoding='ISO-8859-1'?>  
<?xml-stylesheet href="tienda0.xsl" type="text/xsl"?>  
<tienda>  
<nombre>La tiendecilla </nombre>  
<telefono>953 87 12 23 </telefono>  
</tienda>
```



```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet version="1.0"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template  
match='/'>  
<html>  
<head><title>Generado con tienda-html.xsl</title></head>  
<body>  
<h1> <xsl:apply-templates /> </h1>  
</body>  
</html>  
</xsl:template>  
</xsl:stylesheet>
```

# Una vista al XML: Sintaxis

## Declaración de Tipo de Documento

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE lista_de_personas [  
<!ELEMENT lista_de_personas (persona*)>  
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>  
<!ELEMENT nombre (#PCDATA) >  
<!ELEMENT fechanacimiento (#PCDATA) >  
<!ELEMENT sexo (#PCDATA) >  
<!ELEMENT numeroseguridadsocial (#PCDATA)>  
>
```

```
<lista_de_personas>  
  <persona>  
    <nombre>José García</nombre>  
    <fechanacimiento>25/04/1984</fechanacimiento>  
    <sexo>Varón</sexo>  
  </persona>  
</lista_de_personas>
```



**Declaración de Tipo  
de Documento**

# Definición

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE lista_de_personas SYSTEM "docValidarSchema.dtd">
<lista_de_personas>
  <persona>
    <nombre>José García</nombre>
    <fechanacimiento>25/04/1984</fechanacimiento>
    <sexo>Varón</sexo>
  </persona>
</lista_de_personas>
```



**Declaración de Tipo  
de Documento**

*Fichero docValidarSchema.dtd*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT lista_de_personas (persona*)>
<!ELEMENT persona (nombre, fechanacimiento?, sexo?, numeroseguridadsocial?)>
<!ELEMENT nombre (#PCDATA) >
<!ELEMENT fechanacimiento (#PCDATA) >
<!ELEMENT sexo (#PCDATA) >
<!ELEMENT numeroseguridadsocial (#PCDATA)>
```

# Una vista al XML: Sintaxis

## Declaración de Tipo de Documento

La "declaración de tipo de documento" define qué tipo de documento xml estamos creando para ser procesado correctamente. Es decir, describimos la estructura y restricciones que deberá tener los datos del fichero xml para ser válidos respecto a dicha definición.

La declaración de tipo de documento puede estar contenida en el propio fichero xml o en otro fichero, en esta última situación tenemos las siguientes declaraciones:

- ✗ **Usar identificador público (PUBLIC):** que hace referencia a dicha DTD.
- ✗ **Utilizar un Identificador universal de recursos (URI):** precedido de la palabra SYSTEM.

*Ejemplos:*

```
<!DOCTYPE MESAJE SYSTEM "mensaje.dtd">  
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">  
<!DOCTYPE LABEL SYSTEM "http://azuaje.ulpgc.es/dtds/label.dtd">
```

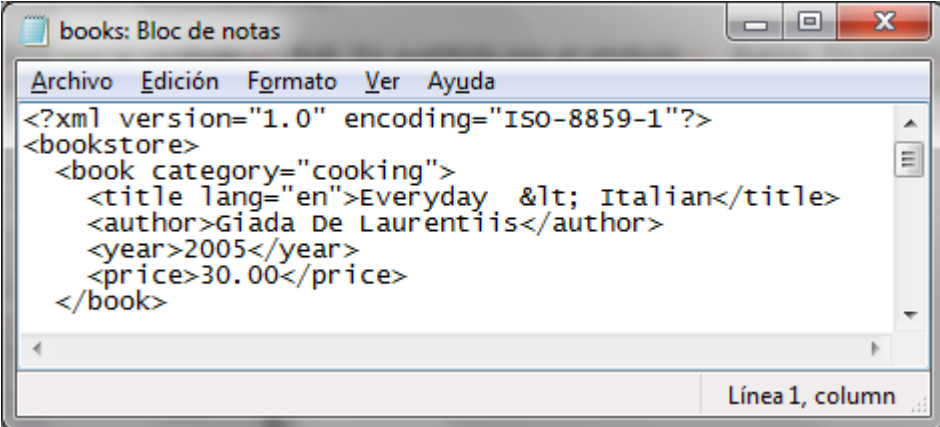
# Una vista al XML: Sintaxis

## Referencias a Entidades: *Entidades internas*

XML proporciona 5 entidades predefinidas que se declaran automáticamente y que están a disposición de todo documento XML:

&lt; Es sustituido por el símbolo < , &gt; Es sustituido por el símbolo > , &amp; Es sustituido por el símbolo & , &apos; Es sustituido por el símbolo ' , &quot; Es sustituido por el símbolo "

```
▼<bookstore>
  ▼<book category="cooking">
    <title lang="en">Everyday &lt; Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  ▼<book category="children">
    <title lang="en">Harry Potter</title>
    <author>J. K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ▼<book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  ▼<book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```



The screenshot shows a Notepad window titled "books: Bloc de notas". The menu bar includes "Archivo", "Edición", "Formato", "Ver", and "Ayuda". The text area contains the following XML code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday &lt; Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
</bookstore>
```

The status bar at the bottom right indicates "Línea 1, column".

# Una vista al XML: Sintaxis

## Comentarios

A veces es conveniente insertar comentarios en el documentos XML, que son ignorados por el procesado de la información y las reproducciones del documento. Los comentarios tienen el mismo formato que los comentarios de HTML. Es decir, comienza por la cadena "<!--" y termina con "-->".

```
<!-- Esto es un comentario -->
```

Se pueden introducir comentarios en cualquier parte del documento salvo dentro de las declaraciones, etiquetas, u otros comentarios.

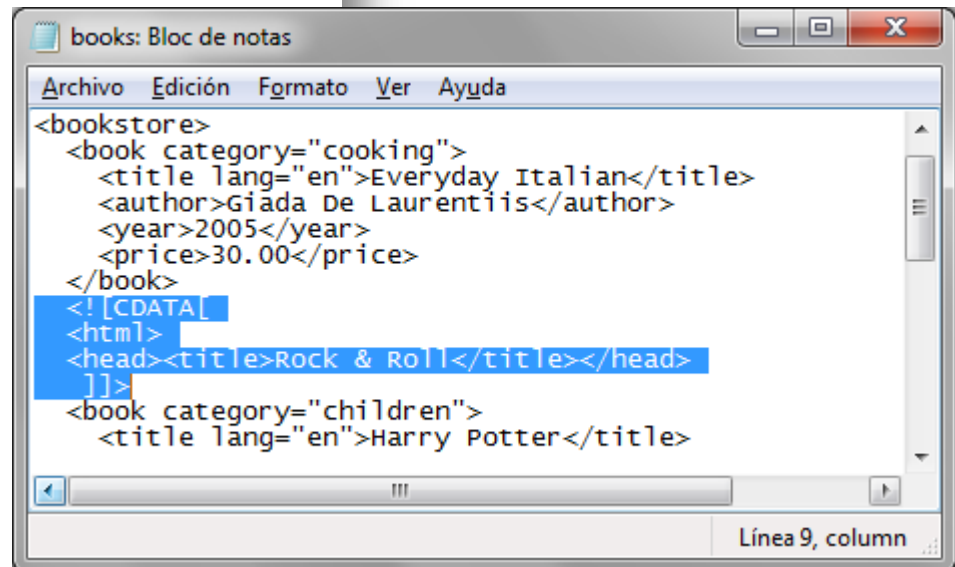
## Secciones CDATA

Existe otra construcción llamada CDATA (Character DATA) en XML, que permite especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML.

Como hemos visto dentro de una sección CDATA podemos poner cualquier cosa, que no será interpretada. Existe una excepción y es la cadena "]]>" con el que termina el bloque CDATA. Esta cadena no puede utilizarse dentro de una sección CDATA.

# Una vista al XML: Sintaxis

```
▼<bookstore>
  ▼<book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <![CDATA[ <html> <head><title>Rock & Roll</title></head> ]]>
  ▼<book category="children">
    <title lang="en">Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  ▼<book category="web">
    <title lang="en">XQuery Kick Start</title>
    <author>James McGovern</author>
    <author>Per Bothner</author>
    <author>Kurt Cagle</author>
    <author>James Linn</author>
    <author>Vaidyanathan Nagarajan</author>
    <year>2003</year>
    <price>49.99</price>
  </book>
  ▼<book category="web" cover="paperback">
    <title lang="en">Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```



```
books: Bloc de notas
Archivo Edición Formato Ver Ayuda
<bookstore>
  <book category="cooking">
    <title lang="en">Everyday Italian</title>
    <author>Giada De Laurentiis</author>
    <year>2005</year>
    <price>30.00</price>
  </book>
  <![CDATA[
  <html>
  <head><title>Rock & Roll</title></head>
  ]]>
  <book category="children">
    <title lang="en">Harry Potter</title>
```

Línea 9, column



# Una vista al XML: Sintaxis

## Espacios de Nombres

Un espacio de nombres XML es una recomendación W3C para proporcionar elementos y atributos con nombre único en una instancia XML. Una instancia XML puede contener nombres de elementos o atributos procedentes de más de un vocabulario XML. Si a cada uno de estos vocabularios se le da un espacio de nombres, se resuelve la ambigüedad existente entre elementos o atributos que se llamen igual. Los nombres de elementos dentro de un espacio de nombres deben ser únicos.

Un ejemplo sería una instancia XML que contuviera referencias a un cliente y a un producto solicitado por éste. Tanto el elemento que representa el cliente como el que representa el producto pueden tener un elemento hijo llamado "numero\_ID". Las referencias al elemento "numero\_ID" podrían ser ambiguas, salvo que los elementos, con igual nombre pero significado distintos, se llevaran a espacios de nombres distintos que los diferenciaran.

# Una vista al XML: Sintaxis

## Espacios de Nombres

```
<?xml version="1.0"?>
<cli:cliente xmlns:cli='http://www.dominio.org/Espacio_de_nombres_XML/cliente'
xmlns:ped='http://www.dominio1.org/Espacio_de_nombres_XML/pedido'>
<cli:numero_ID>1232654</cli:numero_ID>
<cli:nombre>Fulanito de Tal</cli:nombre> <cli:telefono>99999999</cli:telefono>
<ped:pedido>
<ped:numero_ID>6523213</ped:numero_ID>
<ped:articulo>Caja de herramientas</ped:articulo> <ped:precio>187,90</ped:precio>
</ped:pedido>
</cli:cliente>
```

Un espacio de nombres se declara usando el atributo XML reservado `xmlns`, cuyo valor debe ser un identificador uniforme de recurso.

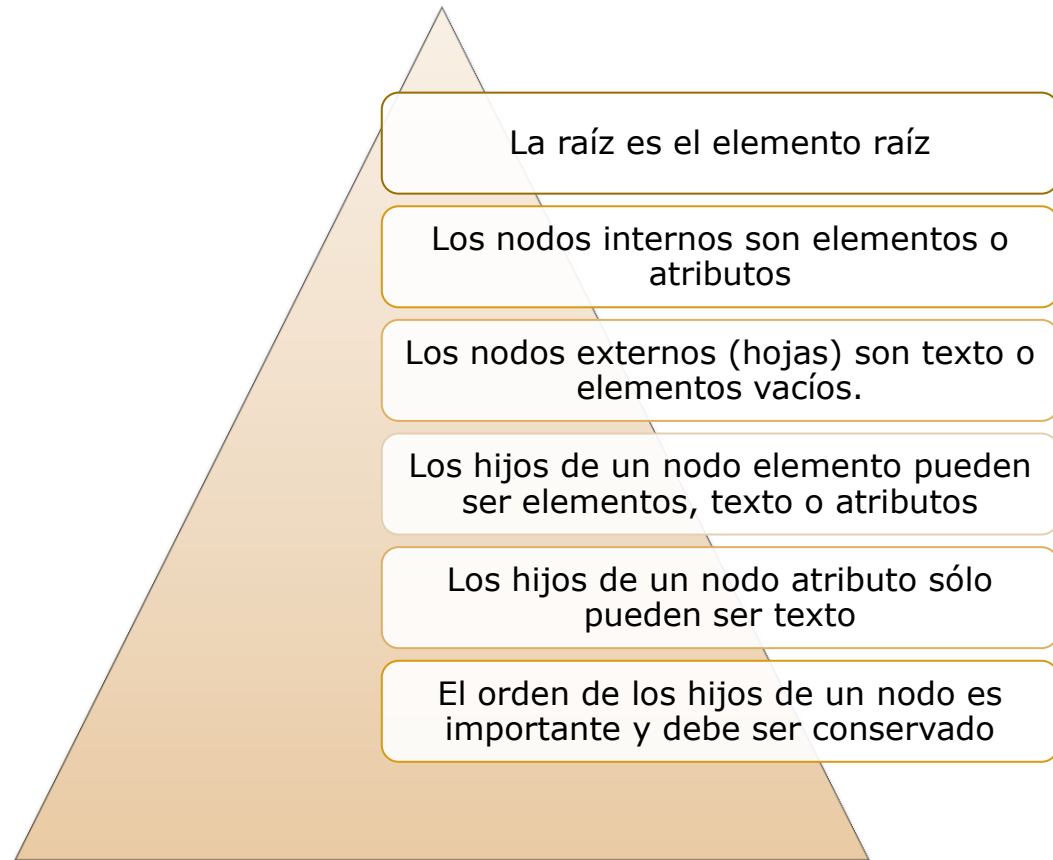
Por ejemplo:

```
xmlns="http://www.w3.org/1999/xhtmll"
```

# Una vista al XML: Sintaxis

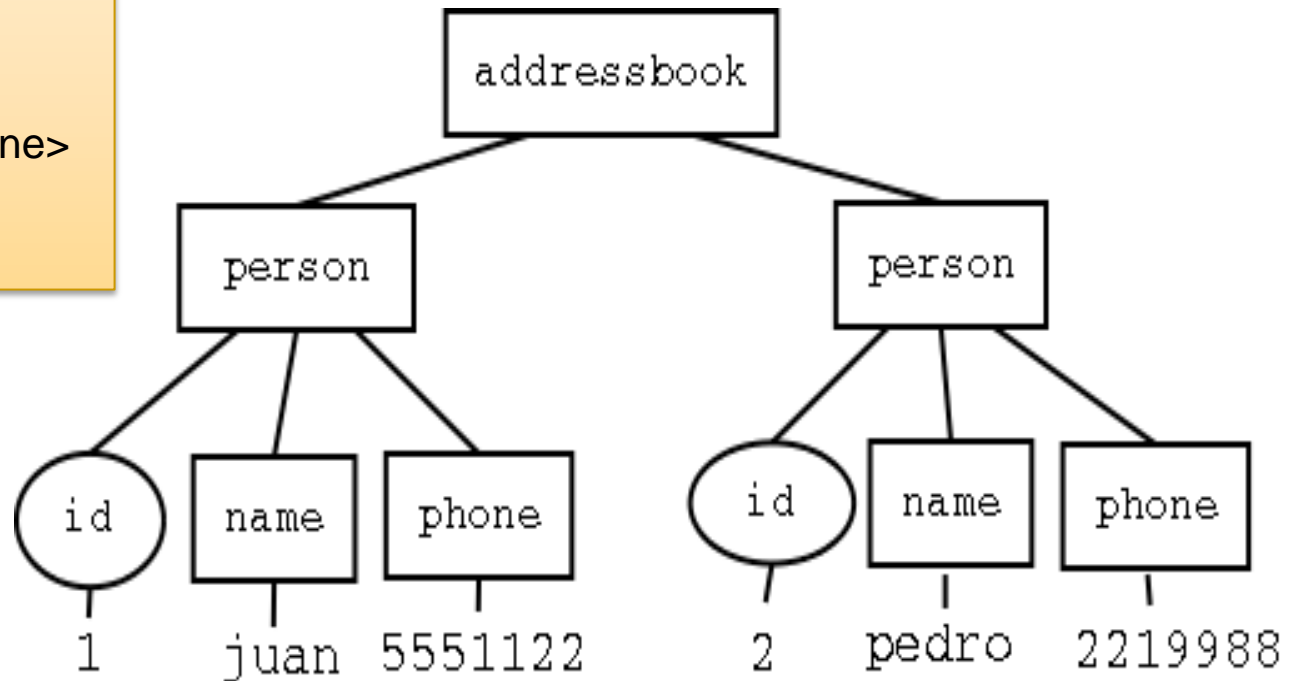
## Árbol descriptivo

Es posible ver todo documento XML como un árbol:



# Una vista al XML: Sintaxis

```
<addressbook>
<person id="1">
<name>juan</name>
<phone>5551122</phone>
</person>
<person id="2">
<name>pedro</name>
<phone>2218899</phone>
</person>
</addressbook>
```



# Una vista al XML: Sintaxis-Resumen

## XML Bien Formados

Los documentos denominados como «bien formados» (del inglés well formed) son aquellos que cumplen con todas las definiciones básicas de formato y pueden, por lo tanto, analizarse correctamente por cualquier analizador sintáctico (parser) que cumpla con la norma.

- ❑ Los documentos han de seguir una estructura estrictamente jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Una etiqueta debe estar correctamente incluida en otra, es decir, las etiquetas deben estar correctamente anidadas. Los elementos con contenido deben estar correctamente cerrados.
- ❑ Los documentos XML sólo permiten un elemento raíz del que todos los demás sean parte, es decir, solo pueden tener un elemento inicial.
- ❑ Los valores atributos en XML siempre deben estar encerrados entre comillas simples o dobles.
- ❑ Las construcciones como etiquetas, referencias de entidad y declaraciones se denominan marcas; son partes del documento que el procesador XML espera entender. El resto del documento entre marcas son los datos «entendibles» por las personas.

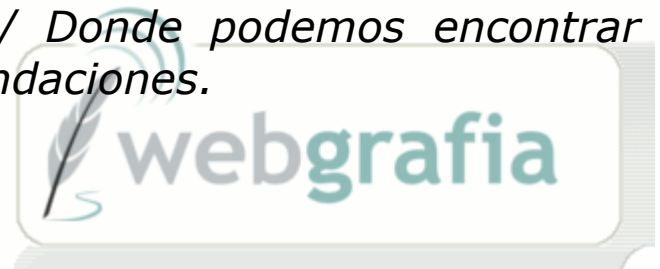
# Una vista al XML: Sintaxis-Resumen

## XML Bien Formados

- ❑ Ningún elemento puede aparecer sin su correspondiente cierre.
- ❑ XML es sensible, por lo que lp y LP se considerarían elementos distintos.
- ❑ Los elementos deben estar anidados correctamente. Así, `<elemento_1><elemento_2> contenido </elemento_2></elemento_1>` es correcto, pero `<elemento_1><elemento_2> contenido </elemento_1></elemento_2>` genera un error en un analizador sintáctico (parser) de XML.
- ❑ Todo documento XML debe tener un elemento raíz que no puede duplicarse, y que es el primero que se abre y el último que se cierra.
- ❑ Los espacios en blanco se preservan.
- ❑ Los comentarios se marcan como en HTML (`<!-- comentario -->`).
- ❑ Los nombres de los elementos pueden contener letras, números y tres signos de puntuación: guión (-), guión bajo (\_) y punto (.).
- ❑ Los nombres de los elementos no pueden empezar por un número o un signo de puntuación (excepto el guión bajo).
- ❑ Los nombres de los elementos no pueden empezar por las letras «xml» (ni cualquier combinación, como XML, Xml, etc.).
- ❑ Los nombres de los elementos no pueden contener espacios.

# Algunos materiales

- ❑ *XML (Curso de Iniciación). David Hunter. ISBN 84-96097-45-5. INFORBOOK'S, S.L.*
- ❑ *Curso de XML. Gregorio Martín e Isabel Martín. ISBN 84-205-4245-8. Prentice Hall.*
- ❑ *<http://www.w3c.es>. Consorcio Internacional W3C. En este sitio podemos encontrar las distintas recomendaciones que el consorcio va creando así como las nuevas versiones.*
- ❑ *<http://www.w3schools.com/> En esta web se pueden encontrar tutoriales e información sobre todos los estándares (recomendaciones) que van apareciendo desde W3C. A la hora de trabajar con XML (y con otras recomendaciones) es una Web imprescindible. Guía rápida y ejemplos de todos los elementos que componen cada una*
- ❑ *XML Imprescindible. Harold, ElliotteRusty u Scott Means, W. Anaya Multimedia-Anaya Interactiva*
- ❑ *<http://www.programacion.net/tutoriales/XML/> Donde podemos encontrar diversos tutoriales sobre XML. de las recomendaciones.*



# Ejercicios

1. ¿Es el XML una versión de HTML, o un lenguaje independiente?
2. ¿En qué formato se escriben los documentos XML?
3. Realiza un resumen/tutorial de XML
4. Busca webreferencias relacionadas con XML
5. ¿Qué ventajas aporta respecto al lenguaje de marcas tradicional?
6. ¿Que mecanismos existen para interpretar un documento XML?
7. ¿Qué es básicamente un DTD?
8. ¿En qué se diferencia esencialmente XML y HTML?
9. Funciones del Consorcio W3C
10. Busca en la www algunas diferencias entre XHTML 1.0 y HTML 4.0
11. Describe alguna diferencia entre xhtml 1.0 y xhtml 2.0
12. ¿Es lo mismo html 5 y xhtml 2.0?
13. Ejemplos de etiquetas que incorpora html 5.0 respecto a html 4.0.
14. Características de XML según recomendación W3C.
15. ¿Qué es básicamente XML?, **Pon un Ejemplo**
16. ¿Qué proporciona la tecnología XSLT?
17. ¿Qué otras tecnologías derivan del universo XML?
18. Elabora el punto **RESUMEN** de este tema.
19. Reglas que debe cumplir un xml bien formado.





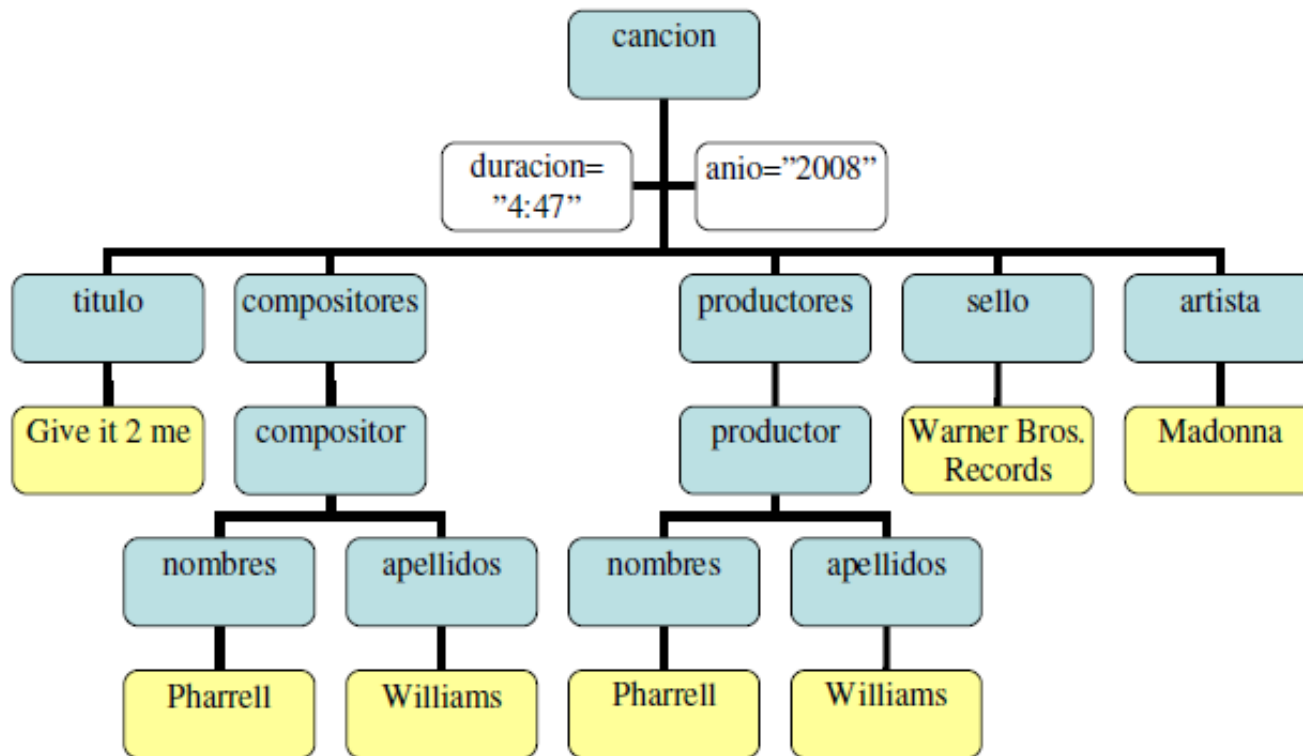
# Ejercicios

Realizar el análisis del siguiente documento XML, generando el árbol que representa su estructura:

```
<?xml version="1.0"?>
  <libro>
    <titulo>El Ingenioso Hidalgo Don Quixote de la Mancha</titulo>
    <autor>Miguel de Cervantes Saavedra</autor>
    <fecha><publicacion>1605</publicacion><edicion>2009</edicion></fecha>
    <localizacion><estanteria>B</estanteria><fila>7</fila></localizacion>
  </libro>
```

# Ejercicios

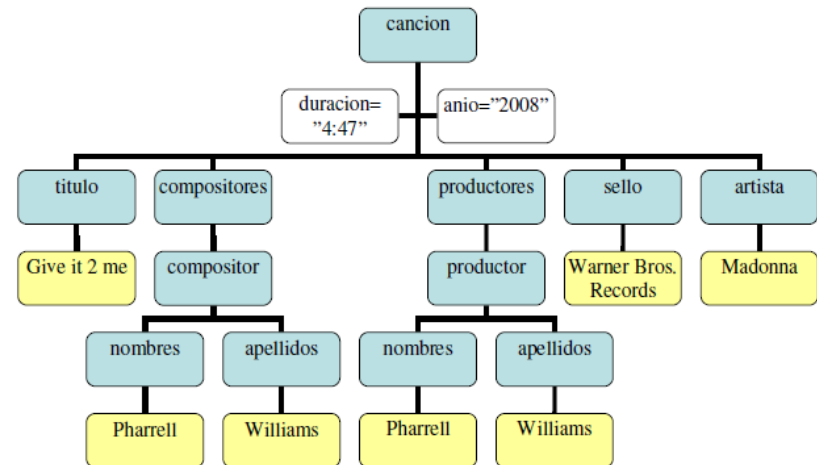
Dado el siguiente árbol descriptivo construye su correspondiente XML:



# Ejercicios

- Amplía el XML realizado en la cuestión anterior para representar la estructura de un CD completo. Para ello debes crear un nuevo elemento *cd* que contenga:

- Un atributo *titulo* que tome como valor el título genérico del CD
- Dos elementos:
  - Uno o varios elementos *cancion* con la estructura que definiste en el apartado anterior.
  - Un elemento *fecha* que sea de carácter EMPTY pero que tenga tres atributos: *dia*, *mes* y *anio* para indicar la fecha de salida al mercado del CD



# Ejercicios

Dado el siguiente XML construye su correspondiente árbol :

```
<Books>
  <Book ISBN="0553212419">
    <title>Sherlock Holmes: Complete Novels...
    <author>Sir Arthur Conan Doyle</author>
  </Book>
  <Book ISBN="0743273567">
    <title>The Great Gatsby</title>
    <author>F. Scott Fitzgerald</author>
  </Book>
  <Book ISBN="0684826976">
    <title>Undaunted Courage</title>
    <author>Stephen E. Ambrose</author>
  </Book>
  <Book ISBN="0743203178">
    <title>Nothing Like It In the World</title>
    <author>Stephen E. Ambrose</author>
  </Book>
</Books>
```

# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Películas (wfd\_01.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<pelicula>
  <titulo>Con faldas y a lo loco</titulo>
  <director>Billy Wilder</director>
</pelicula>
<pelicula>
  <director>Leo McCarey</director>
  <titulo>Sopa de ganso</titulo>
</pelicula>
<autor />barto</autor>
```

# Ejercicios

Determina y corrige el/los error/errores que provoca que el fichero XML **no esté bien formado**

```
<?xml version="1.0" encoding="ISO-8859-1"?><lista_de_personas> <persona
  nacionalidad="español-(Lorca)""> <nombre>José García
  <fechanacimiento>25/04/1984</fechanacimiento>
    <telefono>670201256</telefono>          <telefono>968472898</telefono>
  <sexo>Varón</sexo> </persona> <persona nacionalidad="español">
  <nombre>Juan Antonio</nombre>
  <fechanacimiento>25/04/1984</fechanacimiento>
    <telefono>670201256</telefono>968472898
    <numeroseguridadsocial>2569587412569<numeroseguridadsocial>
  </persona></lista_de_persona>
```

# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Deportistas (wfd\_02.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<deportistas>
<deportista>
  <deporte Atletismo />
<nombre>Jesse Owens</nombre>
<deportista>
<deporte Natación />
<nombre>Mark Spitz</nombre>
</deportista>
</deportistas>
```

# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Texto (wfd\_03.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<texto>
<Titulo>XML explicado a los niños</titulo>
<párrafo>El <abreviatura>XML</abreviatura>define cómo crear lenguajes de
  marcas.</párrafo>
<párrafo>Las marcas se añaden a un documento de texto para añadir
  información.</párrafo>
<http://>www.example.org</http://>
</texto>
```



# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Programas (wfd\_05.xml)

```
<?xml version="1.0" encoding="UTF-8"?> <
programas>
<programa nombre="Firefox" licencia="GPL" licencia="MPL" />
<programa nombre="OpenOffice.org" licencia=LGPL />
<programa nombre="Inkscape" licencia="GPL" />
</programas>
```

# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Mundiales de fútbol (wfd\_06.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<mundiales-de-futbol>
<mundial>
<pais="España" /> <1982 />
</mundial>
</mundiales-de-futbol>
```

# Ejercicios

El siguiente documento no está bien formado. Corrija los errores y construye su árbol descriptivo.

## Medios de transporte (wfd\_07.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<mediosDeTransporte>
<bicicleta velocidad="v<100km/h" />
<patinete velocidad maxima="50 km/h"
</mediosDeTransporte>
```

# Ejercicios

**Elabora en formato XML el fichero de texto que contiene los datos estructurados pertenecientes a un determinado dominio/problema.**

**banco\_lorca.xml:**

- El banco tiene sucursales, cada una identificada por un código.
- Cada sucursal tiene asignadas una serie de cuentas corrientes que también se identifican por un código (distinto para cada cuenta)
- La cuenta tiene asignados uno o varios clientes. Es posible que las operaciones que puede realizar cada uno de estos clientes con la cuenta no sean las mismas.
- Cada cliente, identificado con el NIF, puede tener varias cuentas y, por supuesto, distintos privilegios en cada una de ellas.
- Una cuenta puede tener domiciliaciones asignadas.
- A un cliente se le puede haber concedido uno o varios préstamos. Los préstamos no están asociados a las cuentas, sino a un solo cliente.

# Ejercicios

**Elabora en formato XML el fichero de texto que contiene datos estructurados pertenecientes a un determinado dominio/problema.**

## **Buzón\_correos.xml:**

Contiene la información relativa al conjunto de cartas y paquetes que contiene un buzón de correo (*determina la estructura y el etiquetado*).

## **Libreta\_de\_direcciones:**

Libreta de direcciones con los datos más representativos de los usuarios/clientes. (*determina la estructura y el etiquetado*).

## **MOTORSOL.xml:**

- Los motores están compuestos de piezas que a su vez pueden componerse de otras.
- Nunca un tipo de motor puede ser fabricado por más de una sección, ni esta puede cambiar una vez se le ha asignado la fabricación de un determinado motor.
- Entre los empleados que trabajan en un tipo de motor siempre habrá un responsable, siendo esta información de interés para la empresa. Un empleado puede estar trabajando en más de un tipo de motor.
- Los empleados pueden tener familiares a su cargo.

# Ejercicios

Elabora un XML válido para (*árbol descriptivo de cada solución*):

- una colección *Películas*.
- alumnos del *módulo de LSSGI*
- una colección de *libros*.
- de un *foro*.
- una colección *Clientes*.
- las etapas de un *viaje*.

```
<?xml version="1.0"?><!DOCTYPE books [  
<!ELEMENT title (#PCDATA)>  
<!ELEMENT author (#PCDATA)>  
<!ELEMENT authors (author)+>  
<!ELEMENT subject (#PCDATA)>  
<!ATTLIST subject class CDATA "">  
<!ELEMENT book (title, authors, subject)>  
<!ATTLIST book  
  bookid CDATA #REQUIRED  
  pubdate CDATA #REQUIRED  
>  
<books name="My books">  
  <book bookid="1" pubdate="03/01/2002">  
    <title>Java Web Services</title>  
    <authors>
```

# Ejercicios

*Dado siguiente pedido que ha recibido por correo electrónico una empresa que se dedica a la venta de herramientas para jardinería:*

"Hola, necesito una cortadora de césped para mi jardín de esas que anuncian en oferta, me gustaría que fuera uno de esos modelos eléctricos, pues las de gasolina contamina mucho. Me llamo Roberto Movilla, la cortadora la tendrán que enviar a Albacete, la dirección es Plaza de la Duquesa 12, la recogerá mi esposa que se llama Alicia Abad. Ahora que lo pienso también necesitaré 3 podadoras para los setos. Les paso mi dirección de aquí para lo referente al pago, es General Ricardos 56, aquí en Madrid. Es urgente, por favor, el césped está muy alto."

La fecha del pedido es el 20 de octubre del 99. El empleado que se encarga del pedido ha comprobado algunos datos necesarios: el código postal de la dirección de Albacete es 05020 y el de la de Madrid 28055; también ha consultado el catálogo de productos y ha averiguado que la cortadora vale 148.95 euros y su código de producto es 872-AA, aunque no sabe si es o no eléctrica; una podadora vale 7.98 y su código es 926-FH.

¿Elabora el XML que refleje esta situación?

# Ejercicios

Indica si cada uno de los siguientes es un documento XML bien formado. Si no lo es, explica por qué, y qué se podría cambiar para conseguir que sí lo fuese:

- a) `<?xml version="1.0" encoding="UTF-8"?> <persona> <nombre>Pedro</nombre> <DNI>123456789</DNI> </persona> <persona> <nombre>María</nombre> <DNI>987654321</DNI> </persona>`
- b) `<?xml version="1.0" encoding="ISO-8859-1"?> <texto> Esto es un <negrita>texto</negrita> con formato. </texto>`
- c) `<?xml version="1.0" encoding="ISO-8859-1"?> <texto> Esto es un <negrita><cursiva>texto</negrita> con más</cursiva> formato. </texto>`
- d) `<coche> <marca>Peugeot</marca> <modelo>205</modelo> </coche>`
- e) `<?xml version="1.0" encoding="UTF-8"?> <amigos> <amigo direccion="Avenida de la Felicidad"> Horacio Oliveira</amigo> <amigo direccion="Calle de las Aceitunas"> Pilarín Cañete</amigo> </amigos>`
- f) `<?xml version="1.0" encoding="UTF-8"?> <amigos> <amigo telefono=999999999 telefono=666666666> Horacio Oliveira</amigo> <amigo telefono=933333333>Pilarín Cañete</amigo> </amigos>`
- g) `<?xml version="1.0" encoding="UTF-8"?> <ejercicio> Encuentra la solución de la inequación: <inecuacion>2*x<37</inecuacion> </ejercicio>`



# Ejercicios

Realizar el análisis del siguiente documento XML, generando el árbol que representa su estructura:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
  <FAQ>
    <INFO>
      <TITULO> WINDOWS98? SOLO SI ME OBLIGAN! </TITULO>
      <AUTOR> Fernando Damián Lorenzo García </AUTOR>
      <EMAIL> Fer@yahoo.es </EMAIL>
      <VERSION> 1.0 </VERSION>
      <FECHA> 20.MAYO.99 </FECHA>
    </INFO>
    <PART NO="1">
      <Q NO="1">
        <QTEXT>¿Por qué tengo que instalar W95?</QTEXT>
        <A> A mi también me parece una buena pregunta.</A>
      </Q>
      <Q NO="2">
        <QTEXT>Y de que me vale?</QTEXT>
        <A>Otra ingeniosa pregunta.</A>
      </Q>
    </PART>
  </FAQ>
```

# Sopa de Letras

XFragments

XHTML

DOM

W3C

ISO/ANSI

XML

DTD

XPointer

XLL

XPointer

XQL

Xlink

XSLT

Schema

XSL

CSS

XRef

XPATH

XML-Data

XPath

SGML

Parser

SOAP