



1. Reconocimiento de las características de lenguajes de marcas

Autor	(X) Xerach Casanova
Clase	Lenguajes de Marcas
Fecha	@Dec 7, 2020 3:09 PM

1. [Lenguajes de marcas.](#)
2. [Evolución de los lenguajes de marcas](#)
 - 2.1 [GML \(Generalized Markup Language\).](#)
 - 2.2 [SGML \(Standard Generalized Markup Language\)](#)
 - 2.3 [HTML \(HyperText Markup Language\)](#)
 - 2.4 [XML \(eXtensible Markup Language\)](#)
 - 2.5 [Diferencias entre XML y HTML](#)
 - 2.7. [Etiquetas.](#)
 - 2.8. [Herramientas de edición.](#)
 - [Editores XML](#)
 - [Procesadores XML](#)
3. [XML, estructura y sintaxis.](#)
 - 3.1. [El prólogo](#)
 - [Declaración XML](#)
 - [Declaración del tipo de documento](#)
 - 3.1 [El ejemplar. Los elementos](#)
 - 3.1.2. [Atributos](#)
4. [Documentos XML bien formados](#)
5. [Utilización de espacios de nombres en XML](#)
6. [Mapa conceptual](#)

1. Lenguajes de marcas.

Un lenguaje de marcas es un modo de codificar un documento, donde, junto con el texto, se incorporan etiquetas, marcas o anotaciones con información

adicional. Todo lo cual es relativo a la estructura del texto o su formato de presentación.

Los lenguajes de marcas van definidos en un documento DTD (Document Type Definition). En el cual se establecen las marcas, los elementos utilizados por dicho lenguaje, sus correspondientes etiquetas, atributos, sintaxis y normas de uso.

Se pueden combinar varios tipos diferentes de lenguajes de marcas en un solo documento.

- **De presentación.** Define el formato del texto
- **De procedimientos.** Igual que el de presentación pero interpretado en el orden en que aparece.
- **Descriptivo o semántico.** Se describen las diferentes partes de su estructura sin especificar como se deben representar.

Según el ámbito en el que se empleen se pueden utilizar distintos lenguajes:

- **Lenguajes de marcas de documentación electrónica:** RTF, TeX, Wikibox, Docbook...
- **Lenguajes de marcas de tecnología de internet:** HTML, XHTML, RSS...
- **Otros lenguajes especializados:** MathML, VoiceML, MusicXML...

2. Evolución de los lenguajes de marcas

Paralelamente a los lenguajes de programación para manejar grandes cantidades de datos, los lenguajes de marcas surgen por la necesidad de describir y estructurar la información en los primeros procesadores de textos.

Inicialmente surgen como conjunto de código que los procesadores de textos introducen en los documentos para dirigir el proceso de presentación mediante la impresora, pero estos lenguajes estaban ligados a las características de la máquina o programa, de manera que el "formateador de textos" no podía abstraerse de las características del procesador de textos y trabajar el lenguaje de marcas del documento de forma independiente.

```

1 <times 14><color verde><centrado> Este texto es un ejemplo para mostrar l
2 <color granate><times 10><cursiva>Para realiza este ejemplo se utilizan e
3 Las partes importantes del texto pueden resaltarse usando la
4 <negrita>negrita</negrita>, o el <subrayar>subrayado</subrayar></times 14>

```

Al imprimirlo se obtendría:

Este texto es un ejemplo para mostrar la utilización primitiva de las
marcas

Para realiza este ejemplo se utilizan etiquetas de nuestra invención. Las partes importantes
del texto pueden resaltarse usando la **negrita** , o el subrayado

Posteriormente se añade como medio de presentación la pantalla, de manera que el código desaparecía aunque seguía existiendo para uso interno de las aplicaciones (abstracción). También se incluían otros medios de marcado distintos a la inclusión a mano (combinaciones de teclas, botón...).

Al estar exclusivamente orientado a la presentación de información, pronto surgieron nuevos usos que resolvían gran variedad de necesidades, apareciendo el formato generalizado.

2.1 GML (Generalized Markup Language).

El objetivo de GML es describir documentos de tal modo que el resultado sea independiente de la plataforma y la aplicación utilizada.

Para resolver la falta de formatos de información en los distintos programas, IBM encargó la construcción de un sistema de edición, almacenamiento y búsqueda de documentos.

Para ello crearon un formato estándar que fuera válido para los distintos tipos de documentos legales de la empresa y poder gestionarlos independientemente del departamento o aplicación donde se generó.

2.2 SGML (Standard Generalized Markup Language)

Es la evolución del formato GML que en 1986 dio lugar al estándar ISO 8879.

Lenguaje complejo que requería herramientas y software caro, quedando relegado a grandes aplicaciones industriales.

```
1  <email>
2    <remitente>
3      <persona>
4        <nombre> Pepito </nombre>
5        <apellido> Grillo </apellido>
6      </persona>
7    </remitente>
8    <destinatario>
9      <direccion> pinocho@hotmail.com </direccion>
10   </destinatario>
11   <asunto>¿quedamos?</asunto>
12   <mensaje> Hola, he visto que ponen esta noche la película que querías
13 </email>
```

2.3 HTML (HyperText Markup Language)

Cuando se creó el World Wide Web (www) se encontró la necesidad de organizar, enlazar y compatibilizar gran cantidad de información procedente de diversos sistemas, para ello surge HTML, que era una combinación de:

- ASCII: es el formato que cualquier procesador de textos sencillo puede almacenar permitiendo transferencia de datos entre ordenadores distintos.
- SGML: lenguaje que permite dar estructura al texto aplicando diversos formatos al texto.

HTML es una versión simplificada de SGML. Debido a su comprensión tuvo rápida aceptación, la cual no tuvo SGML, convirtiéndose así en el estándar general para la creación de páginas web.

Desde entonces, las herramientas de software y navegadores que visualizan HTML no han parado de mejorar.

Sus desventajas son:

- No soporta tareas de impresión y diseño.
- No es flexible (etiquetas limitadas)
- No muestra contenido dinámico.
- La estructura y el diseño están mezclados en el mismo documento.

```

1 <html>
2   <head>
3     <title> Ejemplo de código HTML</title>
4   </head>
5   <body>
6     <p></p>
7     <p>
8       <b>23 de julio de 2020</b>
9     </p>
10    <p><b> Bienvenido al modulo de “Lenguajes de Marcas y Sistemas de
11    <p> En este curso aprender&aacute;s, entre otras cosas:<br/>
12    <ul>
13      <li>Las ventajas que ofrece XML </li>
16    </ul>
17    </p>
18  </body>
19 </html>

```

23 de julio de 2020

Bienvenido al modulo de “Lenguajes de Marcas y Sistemas de Gestión de Información”

En este curso aprenderás, entre otras cosas:

- Las ventajas que ofrece XML
- La creación de documentos bien formados
- La creación de DTD

2.4 XML (eXtensible Markup Language)

Para resolver los problemas de HTML en 1998 se establece el estándar XML, un lenguaje de marcas puramente estructural que no incluye ninguna información relativa al diseño.

Las etiquetas indican el significado de los datos y no su formato.

XML es un conjunto de los siguiente estándares:

- XSL eXtensible Style Language, permite definir hojas de estilo y capacidad de transformación de documentos.
- XML Linking Language, incluye Xpath, Xlink y Xpointer y determinan el aspecto sobre los enlaces entre documentos XML.
- XML Namespaces, proveen un contexto al que se aplican las marcas de un XML.

- XML Schemas, permiten definir restricciones a un XML (los más usados son los DTD).

```

1  <?xml version="1.0" encoding="iso-8859-1"?>
2  <!DOCTYPE biblioteca">
3  <biblioteca>
4      <ejemplar tipo Ejem="libro" titulo="XML practico" editorial="Edi
5      <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347";
6      <autor nombre="Sebastien Lecomte"></autor>
7      <autor nombre="Thierry Boulanger"></autor>
8      <autor nombre="Ángel Belinchon Calleja" funcion="traductor"></aut
9      <prestado lector="Pepito Grillo">
10         <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>
11         <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>
12     </prestado>
13 </ejemplar>
14 <ejemplar tipo Ejem="revista" titulo="Todo Linux 101. Virtualización
15     <tipo>
16         <revista>
17             <fecha_publicacion mes="abr" año="2009"></fecha_publicaci
18         </revista>
19     </tipo>
20     <autor nombre="Varios"></autor>
21     <prestado lector="Pedro Picapiedra">
22         <fecha_pres dia="12" mes="ene" año="2010"></fecha_pres>
23     </prestado>
24 </ejemplar>
25 </biblioteca>

```

2.5 Diferencias entre XML y HTML

XML:

- Es un perfil de GSML.
- Especifica cómo deben definirse conjuntos de etiquetas de un mismo documentos.
- Modelo de hiperenlaces complejo.
- El navegador es una plataforma para desarrollo de aplicaciones.
- Fin de la guerra de los navegadores y etiquetas propietarias.

HTML:

- Es una aplicación de SGML

- Aplica un conjunto limitado de etiquetas sobre un único tipo de documento.
- Modelo de hiperenlaces simple.
- El navegador es un visor de páginas.
- El problema de la no compatibilidad y las diferencias entre navegadores alcanza un punto de solución difícil.

Ejemplo fichero con código XML

```

1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <!DOCTYPE libro>
3 <libro>
4   <titulo>XML practico </titulo>
5   <autor>SebastienLecomte</autor>
6   <autor>Thierry Boulanger</autor>
7   <editorial>Ediciones Eni</editorial>
8   <isbn>978-2-7460-4958-1</isbn>
9   <edicion>1</edicion>
10  <paginas>347</paginas>
11 </libro>

```

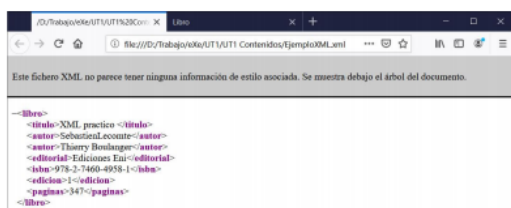
Ejemplo de fichero con código HTML

```

1 <html>
2   <head>
3     <title>Libro</title>
4   </head>
5   <body>
6     <h3>XML practico</h3><br>
7     <p>autores: Sebastien Lecomte,
8       Thierry Boulanger</p>
9     <ul>
10      <li>editorial: Ediciones Eni</li>
11      <li>isbn:978-2-7460-4958-1</li>
12      <li>edicion: 1 </li>
13      <li>paginas: 347</li>
14    </ul>
15  </body>
16 </html>

```

Visualización en Navegador fichero código XML



Visualización en Navegador fichero código HTML



2.7. Etiquetas.

Los lenguajes de marcas utilizan etiquetas especiales intercaladas con texto sin formato, las cuales son interpretadas por intérpretes del lenguaje para procesar el documentos.

Estas se escriben entre ángulos utilizándose dos tipos: etiqueta de inicio < > y etiqueta de fin </ >

Las últimas especificaciones emitidas por W3C indican que siempre vayan en minúsculas.

2.8. Herramientas de edición.

Para trabajar con XML es necesario editar documentos y luego procesarlos.

Editores XML

Basta con utilizar un procesador de textos normal, pero para crear documentos XML complejos se utilizan editores que nos ayuden a crear estructuras y etiquetas.

Algunos incluyen ayuda para la creación de DTD, hojas de estilo CSS o XSL. Un editor gratuito es Amaya.

Procesadores XML

Permiten leer documentos XML y acceder a su contenido y estructura.

Es un conjunto de módulos de software en el que se encuentra el parser, el cual comprueba que el documento cumple las normas establecidas para abrirse.

Pueden ser:

- Validadores (solo trabajan con documentos de tipo válido)
- No validadores (pueden exigir solamente que el documento esté bien formado).

3. XML, estructura y sintaxis.

XML (lenguaje de etiquetas extendido) es un lenguaje de etiquetas creadas por el programador para estructurar y dar forma ordenada a la información. No representa datos por sí mismo.

Ahorra tiempo de desarrollo, permite guardar información de una forma potente y es asimilado por todo tipo de S.O. y dispositivos móviles.

Componentes de documento XML

- Su estructura es de pareja de etiquetas y en árbol.
- El marcado de etiquetas se sitúa entre "<" — ">" — "&" — ";"

- Los comentarios se añaden entre las cadenas "<!-- comentario -->" pero no pueden estar antes del prólogo ni dentro de una etiqueta.
- Se componen de una parte opcional llamada prólogo y otra obligatoria llamada ejemplar.

Sus características son:

- Compatible con protocolos HTTP y URL.
- Todo documento que verifique las reglas XML está conforme con SGML.
- No se requieren conocimientos de programación.
- Fáciles de crear.
- Difusión de documentos asegurada.
- Legible para los humanos.
- Diseño formal y conciso.
- Es extendible y adaptable.
- Orientado a objetos.
- Se compone exclusivamente de datos de marcados y datos carácter entremezclados.

El proceso de creación de documentos XML pasa por varias etapas en las que su éxito depende de la calidad de la anterior:

1. Especificación de requisitos.
2. Diseño de etiquetas.
3. Marcado de documentos.

Se deben crear documentos con estructuras fácilmente actualizables para necesidades futuras.

3.1. El prólogo

El prólogo precede al ejemplar del documento. Facilita el procesamiento de la información y está dividido en

Declaración XML

Ha de ser la primera línea o da error. Es opcional y eso permite que HTML y SGML sea procesado como XML.

Puede tener tres funciones:

1. Declaración de versión XML:
2. Declaración de codificación empleada para representar caracteres.
3. Declaración de la autonomía del documento. Informa si el documento necesita de otro para su interpretación. (sí /no - yes/no)

<?xml versión="1.0" encoding="iso-8859-1" standalone="yes" ?>

123

Los estándar ISO más importantes son:

Estándar ISO	Código de país
UTF-8 (👉 Unicode)	Conjunto de caracteres universal
ISO-8859-1 (Latin-1)	Europa occidental, Latinoamérica
ISO-8859-2 (Latin-2)	Europa central y oriental
ISO-8859-3 (Latin-3)	Sudoeste de Europa
ISO-8859-4 (Latin-4)	Países Escandinavos, Bálticos
ISO-8859-5	Cirílico
ISO-8859-6	Árabe
ISO-8859-7	Griego
ISO-8859-8	Hebreo
ISO-8859-9	Turco
ISO-8859-10	Lapón. Nórdico, esquimal
EUC-JP oder Shift_JIS	Japonés

Declaración del tipo de documento

Define que tipo de documento vamos a crear para que se procesado correctamente.

```
<!DOCTYPE nombre_tipo ....>
```

3.1 El ejemplar. Los elementos

Contiene los datos reales del documento y está formado por elementos anidados.

Los elementos son los distintos bloques de info que permiten definir la estructura del documento. Los elementos pueden estar formados por otros elementos o atributos.

```
1 <?xml version="1.0" encoding="iso-8859-1"?>
2 <!DOCTYPE libro>
3 <libro>
4   <titulo>XML practico </titulo>
5   <autor>Sebastien Lecomte</autor>
6   <autor>Thierry Boulanger</autor>
7   <editorial>Ediciones Eni</editorial>
8   <isbn>978-2-7460-4958-1</isbn>
9   <edicion>1</edicion>
10  <paginas>347</paginas>
11 </libro>
```

El ejemplar es el elemento `<libro>`, que a su vez está compuesto de los elementos `<autor>`, `<editorial>`, `<isbn>`, `<edicion>` y `<paginas>`

El ejemplar es el elemento raíz (root) y todos los datos de un documento XML han de pertenecer a un elemento del mismo.

Los nombres de etiquetas deben ser autodescriptivos.

Las reglas que han de cumplir los documentos XML son:

- Solo un elemento root.
- Todos los elementos tienen una etiqueta de inicio y otra de cierre. En caso de que existan elementos vacíos se empleará `<elemento/>`.
- Los elementos no pueden tener anidados otros elementos sin cerrar.
- Los nombres de las etiquetas de inicio y cierre deben ser idénticos.

- No contener espacios.
- Respetar mayús/minús.
- No puede comenzar por el carácter dos puntos ":".
- No puede comenzar por xml, XML, Xml, etc...
- El contenido de los elementos no puede contener la cadena]] por compatibilidad con SGML. Tampoco se puede utilizar los caracteres < > & " ' sustituibles por:

Carácter	Cadena
>	>
<	<
&	&
"	"
'	'

- Los caracteres especiales £, ©, ®... se debe utilizar &#D (código decimal) &#H (código hexadecimal) correspondiente al carácter en UNICODE. Ejemplo: € es € o €

3.1.2. Atributos

Permiten añadir propiedades a los elementos. No se pueden organizar en jerarquía y no pueden contener otro elemento o atributo. No reflejan estructura lógica. Tampoco se debe utilizar para contener información susceptible de ser dividida.

<pre> 1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?> 2 <!DOCTYPE biblioteca > 3 <biblioteca> 4 <ejemplar tipo_ejem="libro" titulo="XML práctico" editorial="Ediciones Eni"> 5 <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"/></libro> </tipo> 6 <autor nombre="Sebastien Lecomte"/></autor> 7 <autor nombre="Thierry Boulanger"/></autor> 8 <autor nombre="Angel Belinchon Calleja" funcion="traductor"/></autor> 9 <prestado lector="Pepito Grillo"> 10 <fecha_pres dia="13" mes="mar" año="2009"/></fecha_pres> 11 <fecha_devol dia="21" mes="jun" año="2009"/></fecha_devol> 12 </prestado> 13 </ejemplar> 14 </biblioteca> </pre>	<pre> --<biblioteca> --<ejemplar tipo_ejem="libro" titulo="XML práctico" editorial="Ediciones Eni"> --<tipo> <libro isbn="978-2-7460-4958-1" edicion="1" paginas="347"/> --</tipo> <autor nombre="Sebastien Lecomte"/> <autor nombre="Thierry Boulanger"/> <autor nombre="Angel Belinchon Calleja" funcion="traductor"/> --<prestado lector="Pepito Grillo"> <fecha_pres dia="13" mes="mar" año="2009"/> <fecha_devol dia="21" mes="jun" año="2009"/> --</prestado> --</ejemplar> --</biblioteca> </pre>
--	--

Los atributos se definen y dan valor dentro de la etiqueta de inicio o en un elemento vacío, a continuación del nombre del elemento o de la definición de

otro atributo, se separan por espacios. Se definen con comillas simples o dobles y siguen las mismas reglas que los elementos.

4. Documentos XML bien formados

Los documentos XML deben verificar reglas sintácticas que define la recomendación del W3C para el estándar XML, que son:

- Ha de tener definido un prólogo con la declaración XML completa.
- Un único elemento raíz para cada documento, el resto de elementos y contenidos están anidados.
- Hay que cumplir las reglas sintácticas de XML para elementos y atributos.

5. Utilización de espacios de nombres en XML

Permiten utilizar etiquetas idénticas para distintos tipos de información de texto. Dan un nombre único a cada elemento indexándolos según el nombre del vocabulario adecuado asociados a un URI (identificador global único).

De esta forma se evita ambigüedad al juntar dos o más documentos en los que se han definido etiquetas con el mismo nombre. A las etiquetas ambiguas se les precede con un prefijo que determina el contexto al que pertenece la etiqueta seguido de dos puntos:

```
<prefijo:nombre_etiqueta></prefijo:nombre_etiqueta>
```

A esta etiqueta se le denomina "nombre cualificado". El prefijo no puede utilizar espacios ni caracteres especiales y tampoco puede comenzar por un dígito.

Para utilizar un prefijo de un espacio de nombres es necesario declarar el espacio de nombres y asociar un índice con el URI asignado al espacio de nombres con el atributo especial xmlns. Esto se hace entre el prólogo y el ejemplar.

XML de alumnos:

```
1 <?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
2 <!DOCTYPE alumnos>
3 <alumnos>
4   <nombre>Fernando Fernández González</nombre>
5   <nombre>Isabel González Fernández</nombre>
6   <nombre>Ricardo Martínez López</nombre>
7 </alumnos>
```

XML de profesores

```
1 <?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
2 <!DOCTYPE profesores>
3 <profesores>
4   <nombre>Pilar Ruiz Pérez</nombre>
5   <nombre>Tomás Rodríguez Hernández</nombre>
6 </profesores>
```

Uniendo estos dos XML utilizando espacios de nombres quedaría:

```
1 <?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
2 <!DOCTYPE miembros>
3 <alumnos xmlns:alumnos="http://DAM/alumnos">
4 <profesores xmlns:profesores="http://DAM/profesores">
5 <asistentes>
6   <alumnos:nombre>Fernando Fernández González</alumnos:nombre>
7   <alumnos:nombre>Isabel González Fernández</alumnos:nombre>
8   <alumnos:nombre>Ricardo Martínez López</alumnos:nombre>
9   <profesores:nombre>Pilar Ruiz Pérez</profesores:nombre>
10  <profesores:nombre>Tomás Rodríguez Hernández</profesores:nombre>
11 </asistentes>
```

6. Mapa conceptual

