

XML

Baltasar Fernández Manjón



Dpto. de Sistemas Informáticos y Programación,
Universidad Complutense de Madrid
Avda. Complutense s/n, 28040, Madrid, Spain.

<http://bogart.sip.ucm.es/~balta>

XML Lenguaje de marcado extensible

- XML eXtensible Markup Language
- XML es un lenguaje de marcado
 - mas precisamente es un metalenguaje que es una simplificación de SGML para su uso en Internet.
- XML se ha diseñado para describir documentos estructurados y datos (información).
 - Los datos o contenidos se identifican mediante etiquetas o marcas textuales
 - Las marcas son identificadores encerrados entre < y > .
 - P.e. <marca>
 - Documento = contenidos + marcas
- Las etiquetas de XML no están predefinidas como en HTML.
 - Se deben definir unas etiquetas propias según la información que se quiera describir.
 - Las marcas XML indican el significado y contenido de los datos y no como deben presentarse dichos datos.

Ejemplo tabla simple en HTML

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<TABLE border=1>
<TR>
<TH>Curso</TH>
<TH>Departamento</TH>
<TH>Profesor</TH>
<TH>Alumnos</TH>
</TR>
<TR>
<TD>Programación de aplicaciones web</TD>
<TD>SIP</TD>
<TD>Baltasar Fernandez</TD>
<TD>Sistemas
<BR>Gestion</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Atributo de la tabla

Diagram showing the mapping of HTML code to the rendered table structure.

Tabla HTML con mas atributos de presentación

```
<HTML>
<HEAD>
<TITLE>Ejemplo de tabla</TITLE></HEAD>
<BODY>
<TABLE border=1>
<TR>
<TH BGCOLOR=#FFBBBB COLSPAN="4">
  FILA MULTICOLUMNA</TH>
<TH>Curso</TH>
<TH>Departamento</TH>
<TH>Profesor</TH>
<TH>Alumnos</TH>
</TR>
<TR>
<TD>Programación de aplicaciones web</TD>
<TD ALIGN=RIGHT>SIP</TD>
<TD>Baltasar Fernandez</TD>
<TD>Sistemas
<BR>Gestion</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Atributo del título de la tabla

Atributos de la fila de la tabla

Atributo de la celda de la tabla

Diagram showing the mapping of HTML code to the rendered table structure with styling attributes.

Objetivos oficiales de XML

- XML se debe poder utilizar directamente en Internet
- XML debe admitir una gran variedad de aplicaciones
- XML debe ser compatible con SGML
- Debe ser fácil crear programas que procesen documentos XML
- El número de funcionalidades opcionales de XML deberá mantenerse en un mínimo absoluto, preferiblemente cero
- Los documentos XML deberán ser inteligibles para los humanos y razonablemente claros
- El diseño de XML deberá prepararse rápidamente
- El diseño de XML deberá ser formal y conciso
- Los documentos XML deberán ser fáciles de generar
- La concisión en las marcas XML tiene una importancia mínima

© Baltasar Fernández Manjón 5

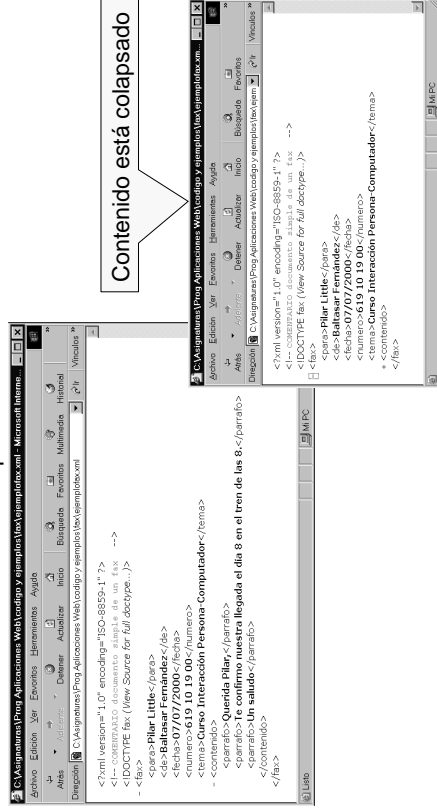
Ejemplo de documento XML

<pre><?xml version="1.0" encoding="ISO-8859-1" ?> <!--COMENTARIO documento simple de un fax; archivo fax.xml --> <!DOCTYPE fax SYSTEM "fax.dtd"> <fax> <para> Pilar Little </para> <de> Baltasar Fernández </de> <fecha>07/07/2000 </fecha> <numero> 619 10 19 00 </numero> <tema> Curso Interacción Persona-Computador </tema> <contenido> <parrafo> Querida Pilar, </parrafo> <parrafo> Te confirmo nuestra llegada el día 8 a las 8. </parrafo> <parrafo> Un saludo </parrafo> </contenido> </fax></pre>	Prólogo Declaración tipo de documento (opcional) Cuerpo del Documento (fax es el elemento documento nodo raíz)
---	---

© Baltasar Fernández Manjón 6

Visualización de un documento XML

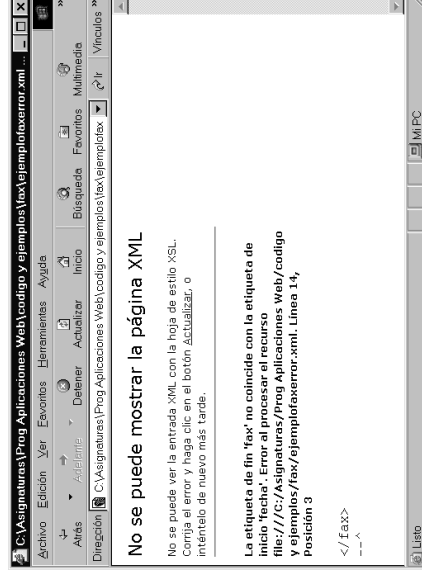
En el Internet Explorer se puede visualizar un documento XML. Se muestra en forma de árbol de modo que se pueden colapsar/expandir cada uno de los elementos compuestos



© Baltasar Fernández Manjón 7

Visualización y comprobación de un documento XML

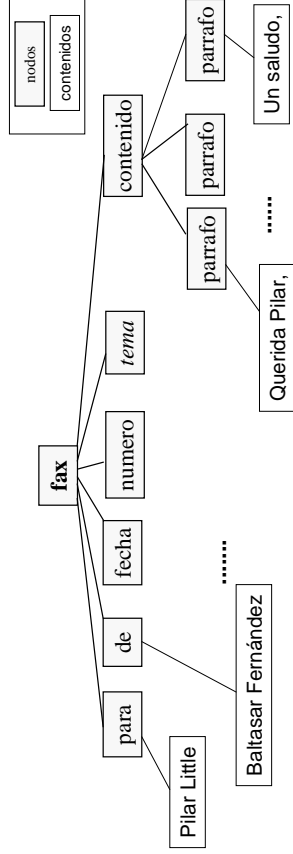
- Además el IE5 incorpora un analizador que si el documento está mal creado muestra un error.
- Por ejemplo si en el fax.xml se omite la etiqueta de cierre del elemento fecha



© Baltasar Fernández Manjón 8

Modelo de datos de un documento XML

- El documento se estructura de forma jerárquica como un árbol de nodos
- Los nodos del árbol son los elementos que conforman el documento
 - Están organizados en niveles denotando composición
- Las hojas del árbol son los contenidos de dichos elementos



©Baltasar Fernández Manjón 9

©Baltasar Fernández Manjón 10

Componentes de un documento XML

- Prólogo
 - Declaración XML
 - Instrucciones de procesamiento
- Contenido
 - Elementos
 - Etiquetas: etiquetas de inicio, etiquetas de fin
 - Etiquetas de elementos vacíos
 - Atributos
 - PCDATA
 - CDATA
 - Espacios en blanco
 - Comentarios
 - Entidades

Partes de un documento XML

- Un documento tiene dos partes principales
 - Prologo
 - Elemento documento (o elemento raíz)
- Prologo
 - No es obligatorio, pero si recomendado
 - Declaración XML
 - información de la versión de XML utilizada
 - conjunto de caracteres utilizado para codificar la información.
 - ¿documento aislado?: `standalone="yes|no"`
 - Declaración del tipo de documento (opcional)
 - Define el tipo y estructura del documento (sintaxis)
 - Una o mas instrucciones de procesamiento (opcional)
 - Proporcionan información que el procesador pasará a la aplicación XML.
 - ♦ P.e. Vincular una hoja de estilo a un documento
- Elemento documento (o elemento raíz)
 - Elemento que contiene al resto de elementos adicionales

©Baltasar Fernández Manjón 11

Elementos

- Componentes básicos de un documento XML
- Encapsulan los contenidos que pueden estar compuestos de:
 - Otros elementos
 - Datos formados por caracteres
 - Referencias a otras entidades
- El identificador de los elementos debe comenzar con una letra y puede contener caracteres de subrayado y de dos puntos pero no espacios en blanco
- Los elementos se delimitan mediante etiquetas formadas por el identificador
- Todos los elementos deben tener las etiquetas de inicio y de fin
 - `<identElemento>contenido del elemento</identElemento>`
- Los elementos vacíos tienen una forma abreviada
 - `<identificadorElementoVacio/>`
- Opcionalmente los elementos pueden contener atributos

©Baltasar Fernández Manjón 12

Contenido o datos formados por caracteres

- El contenido o datos formados por caracteres, es cualquier texto que no son marcas
- Ejemplos:
 - Contenido textual de los elementos
 - Valores de los atributos
 - Un literal de cadena ("dato" o 'dato')
- Los caracteres menor que (" $<$ ") y el ampersand (" $\&$ ") no pueden pertenecer al contenido ya que tienen un significado especial en el mercado
 - Se pueden utilizar empleando secuencias de escape **$\<$** ; y **$\&$** ;

©Baltasar Fernández Manjón 13

Atributos

- Un elemento puede contener atributos que proporcionen información adicional sobre dicho elemento
- En general se utilizan para proporcionar información a una aplicación XML que lo procese
- Los atributos no se consideran parte del contenido de un elemento
- Los atributos se utilizan para asociar pares nombre-valor a los elementos.
 - Los valores de los atributos están formados por cadenas de caracteres
- La especificación de los atributos debe aparecer sólo dentro de las etiquetas de inicio o de las etiquetas de elementos vacíos.

©Baltasar Fernández Manjón 14

Espacios en blanco

- En XML se define como espacio en blanco los siguientes
 - Tabulador
 - Avance de línea
 - Retorno de carro
 - Espacio en blanco
- Un analizador XML debe pasar a la aplicación XML todos los espacios en blanco que aparecen dentro del contenido de un documento
- Un analizador XML debe eliminar todos los espacios en blanco de las etiquetas y de los valores de los atributos
- Los analizadores convierten todos los caracteres de fin de línea en caracteres de avance de línea

©Baltasar Fernández Manjón 15

Referencia a entidades

- Permiten insertar una cadena de caracteres en el contenido de un elemento o en el valor de un atributo
- Hay cinco entidades predefinidas en XML:
 - $\<$; ($<$)
 - $\&$; ($\&$)
 - $\>$; ($>$)
 - $\'$; ($'$)
 - $\"$; ($"$)
- También es posible crear entidades definidas por el usuario

©Baltasar Fernández Manjón 16

Documentos bien formados

- Documentos que cumplen las reglas básicas de XML de modo que pueden ser procesados por un programa
- Reglas básicas
 - El documento debe tener exactamente un elemento de nivel superior (elemento documento o elemento raíz)
 - La primera marca del documento es la marca del elemento raíz
 - Los elementos deben estar adecuadamente anidados
 - Cada elemento debe tener una marca de inicio y una marca de fin
 - Los elementos vacíos pueden indicarse con una marca especial
 - El nombre del tipo de elemento de una marca de inicio debe corresponder exactamente con su marca de fin correspondiente
 - En los nombres de los tipos de elementos se distingue entre mayúsculas y minúsculas
 - No puede aparecer un atributo más de una vez, en un mismo elemento
 - El valor de los atributos debe ir entre comillas

17

©Baltasar Fernández Manjón

Documentos XML válidos

- Un documento XML bien formado que además sigue las reglas especificadas en una Definición de Tipo de Documento (DTD)
 - La DTD define la estructura del documento y los elementos que pueden componer dicho documento
- La DTD sirve para describir los datos. XML está diseñado para que un documento conjuntamente con su DTD sea autodescriptivo y se pueda validar automáticamente su corrección
- Una DTD puede declararse dentro de un documento o mediante una referencia externa.
 - Declaración interna al documento:
 - <!DOCTYPE elemento-raíz [declaraciones de elementos]>
 - Es mejor realizar una declaración externa

18

©Baltasar Fernández Manjón

Declaración de tipo de documento

- Declaración mediante referencia externa
 - <!DOCTYPE elemento-raíz SYSTEM "URI" [declaraciones internas de elementos]>
 - Si existen declaraciones comunes dentro de un documento y en la referencia a la DTD externa tienen prioridad las declaraciones internas
- SYSTEM seguido por un URI (Universal Resource Identifier)
 - URL
 - Camino relativo (path)
- En vez de SYSTEM puede aparecer PUBLIC seguido por [declaraciones internas de elementos]
 - <!DOCTYPE elemento-raíz PUBLIC "identificador" "URI" [declaraciones internas de elementos]>

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<!DOCTYPE fax SYSTEM "fax.dtd">
```

19

©Baltasar Fernández Manjón

Declaraciones básicas en una DTD

- Las declaraciones en una DTD tienen la siguiente forma
 - <!keyword parámetro1 parámetro2 ... parámetroN>
- Hay 4 palabras reservadas básicas
 - ELEMENT
 - Declara un nombre de tipo de elemento y sus posibles subelementos
 - ATTLIST
 - Declara los nombres de los atributos de un elemento, así como sus posibles valores y/o valor por defecto
 - ENTITY
 - Declara referencias a caracteres especiales o a bloques de texto (similar a un #define) o también a contenido repetitivo que puede estar contenido en un recurso externo (similar a un #include).
 - NOTATION
 - Declara contenido no-XML externo (p.e. Imágenes) y la aplicación externa que gestiona dicho contenido

20

©Baltasar Fernández Manjón

DTD XML simple para un fax

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!--
  DTD simple para un fax.
-->
<ELEMENT fax (para, de, fecha, numero, tema?, contenido)>
<ELEMENT para (#PCDATA)>
<ELEMENT de (#PCDATA)>
<ELEMENT fecha (#PCDATA)>
<ELEMENT numero (#PCDATA)>
<ELEMENT tema (#PCDATA)>
<ELEMENT contenido (parrato+)>
<ELEMENT parrato (#PCDATA)>
```

Archivo fax.dtd

Declaración de elementos en una DTD

- La declaración de un elemento debe tener alguna de las dos siguientes formas
 - <!ELEMENT nombre_elemento categoría_contenido>
 - <!ELEMENT nombre_elemento (modelo_contenido) cardinalidad>
- Categorías de contenido para elementos
 - ANY
 - Puede contener cualquier XML bien formado
 - EMPTY
 - No puede contener nada (salvo atributos)
 - Sólo texto: PCDATA
 - No puede contener a otros subelementos
 - Sólo elementos
 - Contiene únicamente elementos hijos (o subelementos)
 - Contenido mixto
 - Puede contener tanto texto como otros elementos

ejemplo

Declaración:

<!ELEMENT br EMPTY>

VÁLIDO:

```
<br />
<br></br>
<br><item id="x" /></br>
<br></br>
<br>
</br>
```

Modelos de contenido

- Excepto en las categorías ANY o EMPTY es necesario proporcionar un modelo de contenido
- Normalmente un modelo de contenido es una lista de nombres de elementos o el identificador PCDATA encerrados entre paréntesis
 - Pueden aparecer mas paréntesis con propósito de agrupar otros elementos
- Dentro de los modelos de contenido pueden aparecer dos tipos de listas
 - Listas de secuencia
 - Los elementos hijos aparecen en orden separados por comas
 - Listas de elección
 - Sólo puede aparecer uno de los elementos especificados.
 - Los elementos se separan mediante barras verticales

Contenido textual - PCDATA

- Sólo puede incorporar contenido textual y referencia a entidades
- No puede incluir a otros subelementos

Declaración:

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

Fragmento de documento válido:

```
<parrrafo> Querida Pilar, </parrrafo>  
<parrrafo> Te confirmo nuestra llegada el día 8 a las 8.  
</parrrafo>  
<parrrafo> Un saludo </parrrafo>  
<parrrafo> puede incluir referencias a entidades como &amp; </parrrafo>
```

Contenido de subelementos

- Sólo puede contener a otros elementos
- No puede contener texto fuera de los elementos hijos o subelementos

Declaración:

```
<!ELEMENT fax (para, de, fecha)>
```

Fragmento de documento válido:

```
<fax>  
<para> Pilar Little </para>  
<de> Baltasar Fernández </de>  
<fecha>07/07/2000 </fecha>  
</fax>
```

Fragmento de documento no válido:

```
<fax>  
este contenido textual no es  
válido  
<para> Pilar Little </para>  
<de> Baltasar Fernández  
</de>  
<fecha>07/07/2000  
</fecha>  
</fax>
```

Contenido mixto

- Puede contener tanto texto como otros elementos
- Siempre se especifica utilizando una lista de elección
- Siempre que aparezca la palabra clave PCDATA debe ser el primer ítem del modelo de contenido
- En el modelo de contenido mixto no se puede restringir el número de apariciones de los subelementos

```
<!ELEMENT foo (#PCDATA | bar | otro) * >
```



```
<!ELEMENT foo (bar | #PCDATA | otro) * >
```



Operadores de cardinalidad

- Los operadores de cardinalidad definen cuantos elementos hijo pueden aparecer en un modelo de contenido
- Operadores
 - Ninguno
 - La ausencia de operador indica que es necesaria y sólo se permite una y sólo una ocurrencia del elemento
 - ?
 - Cero o una instancia del elemento (denota opcionalidad)
 - *
 - Cero o mas instancias (opcional y repetible)
 - +
 - Una o mas instancias (obligatorio y repetible)

Ejemplo

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE XXX [
  <!ELEMENT XXX (AAA+, BBB+)>
  <!ELEMENT AAA (BBB | CCC) >
  <!ELEMENT BBB (#PCDATA | CCC) *>
  <!ELEMENT CCC (#PCDATA)>|>
<XXX>
<AAA>
<CCC>un único elemento.</CCC>
</AAA>
<AAA>
<BBB><CCC/><CCC/><CCC/></BBB>
</AAA>
<BBB/>
<BBB>
Esta es <CCC/> una combinación <CCC/> de <CCC>elementos CCC </CCC> y
texto <CCC/>.
</BBB>
<BBB> sólo texto </BBB>
</XXX>
```

Declaración de DTD
interna al documento