DOCUMENTOS XML

Apuntes básicos del Tema 4

Contenidos:

1. El lenguaje de marcas XML.
2. XML, HTML, XHTML
3. Estructura del documento XML
4. Codificación de los documentos
5. Elementos
6. Atributos
7. Caracteres especiales
8. Comentarios
9. Documentos bien formados

1. El lenguaje de marcas XML

Hemos visto en el tema de introducción a la asignatura que **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. Pero desgraciadamente SGML, es un lenguaje tan complicado que no resulta adecuado para el intercambio de datos en la web.

Una simplificación de este amplio lenguaje se recoge en las especificaciones de XML. Es decir, **XML es un subconjunto de SGML**, con los mismos objetivos, pero sin el grado de complejidad del lenguaje original. XML, fue diseñado para que sea totalmente compatible con SGML, lo que implica que cualquier documento que sigue las reglas de sintaxis de XML es también por definición un documento SGML

XML, son las siglas en inglés de ***eXtensible Markup Language.***

Es importante tener en cuenta que XML no es un lenguaje, sino un estándar para crear lenguajes que cumplen los criterios XML. En otras palabras, **XML describe una sintaxis que se utiliza para crear nuestros propios lenguajes**. Por esa razón se dice que en realidad es un ***metalenguaje***.

Por ejemplo, un fichero con un documento XML muy sencillo podría contener la siguiente información :

```
<?xml version="1.0" encoding="UTF-8"?>
<agenda>
    <nombre_completo>
        <nombre>
            Pepe
        </nombre>
        <apellido1>
            Gonzalez
        </apellido1>
        <apellido2>
            Ribera
        </apellido2>
    </nombre_completo>
    <apodo>
        Pepito Grillo
    </apodo>
</agenda>
```

Entendemos así la razón por la que a los lenguajes de marcas (SGML y XML) se les llama *autodescriptivos,* ya que vemos inmediatamente que la información que queremos tratar está relacionada con el nombre de una persona. Y además la presentamos desglosada de una forma específica y deliberadamente prevista. Los nombres que hemos asignado a la información sigue un formato de etiquetas, en las que siempre tenemos la de inicio y la de fin, y entre ellas el contenido que le corresponde.

Los nombres que hemos asignado a las etiquetas son totalmente arbitrarios, nos los hemos inventado nosotros, y podrían haber sido cualesquiera otros, pero en XML siempre se recomiendo usar nombres significativos y autodescriptivos.

Pero la descripción de los elementos no ha sido arbitraria, ha seguido unas normas determinadas, como la marca de etiqueta inicial y la final, dentro de los caracteres (< >) correspondientes, o el respeto al anidamiento de las distintas etiquetas. Eso es por que sigue un estándar, que garantiza la compatibilidad entre los que lo siguen. Un ejemplo de estándar conocido por todos es el que se sigue en el conocido HTML.

Podemos preguntarnos ¿Quién crea los estándares?

> En 1994 se formó W3C (World Wide Web Consortium), con el objetivo, tal y como se expresa en su sitio http: //www.w3c.org, “de lograr que la web desarrolle todo su potencial por medio de la creación de protocolos comunes que promuevan su evolución y que aseguren la interoperabilidad”. W3C produce recomendaciones que construyen la estructura básica de la web. W3C las denomina **recomendaciones**, en lugar de estándares, porque los desarrolladores y fabricantes tienen la libertad de seguir esas recomendaciones que brindan esa interoperabilidad.
>
> Su contribución más conocida es, la recomendación HTML. Cuando un navegador web señala que se cumple con la versión 3.2 o 4.01 de las recomendaciones HTML, se están refiriendo a la recomendación desarrollada bajo la autoridad de W3C.
>
> La razón por la que las especificaciones de W3C se implementan de manera tan extendida se debe a que la creación de estos estándares es un proceso abierto: cualquier empresa o individuo puede ser miembro de W3C, y ser miembro significa que esas empresas o individuos pueden formar parte del proceso de creación de los estándares.

En realidad, un documento XML no es más que un fichero de texto donde la información viene etiquetada.

Por tanto, estos documentos pueden ser leídos por personas (en un bloc de notas o en un navegador) pero también por sistemas que sacan información del documento XML y por ello estará formado por una serie de elementos definidos en las **recomendaciones** XML de W3C ya que para que lo lea un ordenador el fichero deberá cumplir una serie de reglas para que la aplicación que lo lea sea capaz de interpretarlo. Los documentos XML que cumplen estas reglas se dice que son **documentos bien formados** (más adelante especificaremos las reglas paraque sea así considerado)

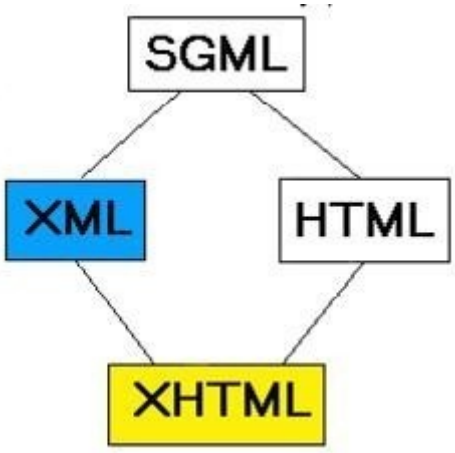
Como conclusión importante, distingamos el concepto de ***Documento XML*** y el del propio ***Estándar XML*** que será el conjunto de reglas para definir etiquetas y otras especificaciones para construir los documentos. Por eso se define en ocasiones a XML como un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas.

2. XML, HTML, XHTML

HTML (HyperText Markup Language) es un lenguaje de marcas muy sencillo que permite especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas, etc.) y los efectos que se visualizan (cursiva, negrita) y dejando que la presentación final se realice por medio del navegador.

El lenguaje **HTML** es un subconjunto de SGML, y como hemos visto, XML es un metalenguaje que permite definir lenguajes que cumplen las recomendaciones SGML. **XHTML** (eXtensible Hypertex Markup Language) puede considerarse una nueva versión de HTML que cumple totalmente las especificaciones XML y que se está convirtiendo en el nuevo estándar de páginas Web.

Podríamos ver su procedencia como:



En apariencia XML se parece mucho a HTML, pero es crucial entender algunas diferencias:

- XML es extensible, lo que significa que este lenguaje puede ser extendido y adaptado para responder a necesidades diferentes. XML es un metalenguaje

de marcado sin una lista fija de etiquetas, pues estas se crean en función de las necesidades del proyecto.

- XML tiene que estar bien formado (“well-formed”). Para que ello se cumpla, debe cumplir con una serie de normas bastante estrictas; esto permite a los procesadores XML parsear y poder validar cualquier documento. De hecho, si un documento no está bien formado, será rechazado por cualquier procesador y por tanto inservible.
- XML, como hemos dicho, se centra en la descripción semántica de los fragmentos de texto, mientras que HTML se ocupa solo de la presentación. XML se preocupa del qué son las cosas, y no del cómo se ven; separa, en definitiva, el contenido de la presentación.
- XML puede ser validado a través de un esquema específico, un aspecto que veremos más adelante. Los documentos XML, además de las reglas intrínsecas del mismo lenguaje, pueden depender de un esquema que determine qué elementos se pueden utilizar y de qué manera se pueden anidar, qué tipos de atributos pueden llevar, etc. En el caso de que el documento XML cumpla los requisitos del esquema, será “válido”.

Resumiendo: ¿Qué es XML?

XML es un lenguaje de marcas que ofrece un formato para la descripción de datos estructurados, el cual conserva todas las propiedades importantes de SGML. Es decir, XML es un **metalenguaje**, dado que con él podemos definir nuestro propio lenguaje y, a diferencia del HTML, que se centra en la representación de la información, XML se centra en la información en si misma.

La particularidad más importante del XML es que no posee etiquetas prefijadas con anterioridad, ya que es el propio diseñador el que las crea a su antojo, dependiendo del contenido del documento.

De esta forma, los documentos XML con información sobre libros deberían tener etiquetas como <AUTOR>, <EDITORIAL>, <Nº_DE_PÁGINAS>, <PRECIO>, etc., mientras que los documentos XML relacionados con educación incluyen etiquetas del tipo de <ASIGNATURA>, <ALUMNO>, <CURSO>, <NOTA>, etc.

Al ser XML un metalenguaje es un lenguaje para definir lenguajes. Los elementos que lo componen pueden dar información sobre lo que contienen y la estructura del documento nos da información de como se relacionan unos datos con otros.

3- Estructura del documento XML

La estructura general de un documento XML está formada por dos partes:

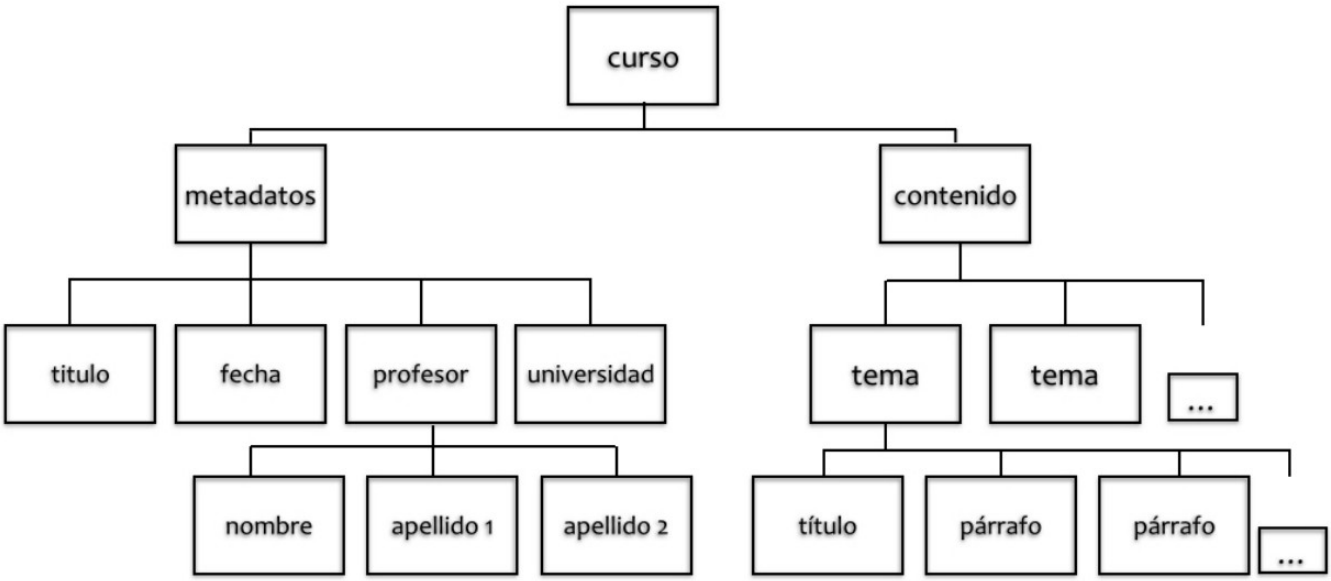
- Prólogo (opcional): contiene una secuencia de instrucciones de procesamiento y/o declaración del tipo de documento. Se puede dividir en dos partes:
  - Declaración XML. Establece la versión de xml, el tipo de codificación y si es un documento autónomo.
  - Declaración de tipo de documento. Establece el tipo de documento que es.
- Cuerpo: es el contenido de información del documento, organizado como un árbol único de elementos marcados.

Un documento XML está formado por una mezcla de datos e información de etiquetado sobre los mismos. La información de etiquetado se corresponde con el texto que aparece encerrado entre los caracteres ‘<’ y ‘>’.

Es importante resaltar que XML **distingue entre mayúsculas y minúsculas**, tanto en los datos como en el etiquetado. Por ejemplo, las etiquetas <nombre> y <NOMBRE> son completamente diferentes.

Todo documento XML tiene un solo elemento raíz (en inglés, “root”), el único que no depende de ningún otro elemento y que contiene todos los otros.

Ejemplo: un documento XML



Como vemos, la estructura arbórea consta de un solo elemento raíz (“curso”), del que descienden todos los otros elementos. Estos elementos pueden constituirse en lo que llamamos “nodos” (en inglés “nodes”) que pueden ser: a) un subárbol: por ejemplo, metadatos con sus cuatro elementos descendientes titulo, fecha, profesor, universidad; b) un simple elemento, por ejemplo, el nodo titulo; c) la misma secuencia de caracteres que encontramos como contenido al interior de los elementos. Es importante conocer esta nomenclatura y utilizar este vocabulario para no dar lugar a equívocos.

Veremos y analizaremos la estructura continuando y ampliando el ejemplo 1:

anterior:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE  agenda SYSTEM "agenda.dtd">
<agenda>
    <nombre_completo>
        <nombre>
            Pepe
        </nombre>
        <apellido1>
            Gonzalez
        </apellido1>
        <apellido2>
            Ribera
        </apellido2>
    </nombre_completo>
    <apodo>
        Pepito Grillo
    </apodo>
</agenda>
```

(ejemplo 1 ampliado)

- En la primera línea tenemos el **prólogo.** El prólogo añade información sobre el documento. En concreto, declara que el documento es un documento XML e incluye información sobre la **versión** de XML utilizada para escribirlo (la 1.0).

  Además, puede incluir información sobre el tipo de codificación de caracteres utilizado en el documento, si es autónomo (contiene en sí mismo toda la información necesaria para procesarlo) o no y el tipo al que se ajusta el documento. Si se incluye, el prólogo debe preceder al ejemplar del documento.

  Aunque el prólogo es opcional, su inclusión es muy recomendable ya que facilita un procesado fiable y robusto de la información contenida en el ejemplar

- En la segunda línea tenemos la declaración de tipo de documento (DTD). Este tipo de declaración sirve para validar los documentos XML. También es opcional, pero veremos que lo necesitamos para garantizar una determinada estructura predeterminada. En el próximo tema lo veremos ampliamente.
- El resto del documento forma el cuerpo de él, y está formado por elementos, delimitados por etiquetas de inicio y fin y un contenido,
- El documento debe tener siempre un elemento raíz que contiene a todos los demás.

4-. Codificación de los documentos

XML usa por defecto el estándar de representación UTF-8. Este estándar está en continua revisión y contiene más de setenta mil caracteres diferentes.

Esta codificación emplea un numero variable de bytes para la representación de los caracteres. Representa los caracteres ASCII con un solo byte, lo que asegura compatibilidad y un tamaño reducido en los documentos que contengan sólo información tipo ASCII. Los caracteres del castellano no incluidos en ASCII (vocales acentuadas, eñe, etc.) ocupan 2 bytes. Debido a ello, en muchos entornos de edición en castellano se usa la codificación ISO-8859-1 o Latín 1. Se trata de una extensión de ASCII, que codifica los caracteres especiales de las lenguas europeas occidentales en un solo byte.

En cualquier caso, siempre se puede incluir cualquier carácter, aunque sea indirectamente, mediante el mecanismo denominado ***referencias a carácter***. Una referencia a carácter especifica el valor numérico del carácter deseado.

El formato de la referencia a carácter, en el caso que el valor numérico sea decimal es:

- &#código_decimal;

Y si el número es hexadecimal

- &#xcódigo_hexadecimal;

URL códigos entidades:

https://es.wikipedia.org/wiki/Anexo:Referencias_a_entidades_de_caracteres_XML_y_HTML

5. Elementos

Un documento XML está formado por una jerarquía de elementos a partir de un elemento raíz único. Se podría definir el elemento como el componente básico de un documento.

Un elemento consta de una etiqueta de inicio, un contenido y una etiqueta de fin. Pueden tener atributos.

Cada dato de un documento tiene que formar parte del contenido de algún elemento. La sintaxis de la etiqueta de inicio es <NombreElemento> y la de fin es </NombreElemento>.

El nombre de un elemento debe empezar por una letra o el carácter subrayado, seguido por letras, dígitos, el carácter punto, el carácter guión o el carácter subrayado. Además, el nombre no deberá comenzar por la cadena XML, ya sea en mayúsculas o en minúsculas. Estas normas también se aplican al nombre de los atributos (si los lleva).

Los elementos se pueden clasificar según el contenido en:

- Elementos que solo contienen texto.
- Elementos que contienen otros elementos.
- Elementos vacíos
- Elementos que contiene texto y otros elementos.

Lo más usual y que veremos en los primeros ejemplos serán los dos primeros, es decir, los que solo contienen texto u otros elementos

6. Atributos

Un atributo es una información complementaria asociada a un elemento. Aparece en la etiqueta de inicio y cada elemento puede tener una lista de atributos asociada, en la que el orden es intranscendente pero no pueden aparecer atributos repetidos. Un atributo consiste en una pareja de nombre y valor, donde el valor debe aparecer encerrado entre comillas simples o dobles, siguiendo la siguiente sintaxis:

<elemento **atributo1**=”valor1” **atributo2**=”valor2”>

Por ejemplo:

```
<peso unidad=”gramos” precision=”0,01”>5,73</peso>
```

Al diseñar un documento XML puede surgir la duda de cuándo usar elementos y cuándo atributos.

El ejemplo anterior, en otro contexto, quizás se hubiese pensado formarlo solo por parte de elementos, es decir, con una estructura similar a:

```
<ingrediente>
        <tipo> tomates</tipo>
        <peso> 5,73 </peso>
        <unidad> gramos </unidad>
        <precision> 0,01 </precision>
</ingrediente>
```

En este caso está claro ya que parece lógico que cada elemento *peso* incluya como atributos la unidad de medida y la precisión, en otros casos es más dudoso.

Aunque no se puede dar un criterio válido para todos los casos si podemos dar algunas pautas que ayuden a tomar una decisión:

- Si la información tiene una estructura interna debe ser un elemento
- Si contiene una gran cantidad de información, parece más adecuado, un elemento
- Los atributos no pueden contener múltiples valores (los elementos sí)
- Los atributos no son fácilmente expandibles (para futuros cambios)
- Los mecanismos de procesamiento y presentación de documentos, permiten tener mejor control sobre los elementos. Por tanto aquella información que tenga un procesamiento o presentación complejos debe ser un elemento.

Como consejo final, en caso de duda, se debe utilizar un elemento.

En este ejemplo **date** es usado como atributo:

```
<note date="2008-01-10">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

En este ejemplo **<date>** ese emplea como elemento:

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

En este ejemplo los datos de <date> se expanden con sus propios elementos:

```
<note>
  <date>
    <year>2008</year>
    <month>01</month>
    <day>10</day>
  </date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

De todas maneras se recuerda que no hay reglas sobre cuándo usar atributos o cuándo usar elementos en XML.

7. Caracteres especiales

Si como contenido de un elemento tuviésemos que utilizar caracteres especiales como por ejemplo para plantear una expresión lógica de comparación:

10 < 20 & 50 >5

al crear elemento tendríamos

<comparacion**>10 < 20 & 50 >5<**/comparacion**>**

Si visualizamos el texto anterior con un navegador nos dará un error ya que al encontrarse con el carácter '<' espera una etiqueta y no un carácter en blanco.

Existen varios caracteres que tienen un significado especial (<, &, >, ' y "). ¿Qué ocurre si se requiere incluir uno de estos caracteres dentro del documento XML? En el caso que pueda existir algún tipo de ambigüedad, será necesario establecer un mecanismo que permita distinguir si el carácter debe interpretarse como especial o no.

Una primera manera de indicar que no se interprete como especiales ninguno de los caracteres de un texto es mediante la sección CDATA, que se inicia con la cadena <![CDATA[y termina con]]>, y que hace que no se interprete como etiquetas el texto que contiene.

Por lo tanto, para incluir la expresión anterior en una etiqueta tendríamos:

<comparacion><**![CDATA[<10 < 20 & 50 >5>]]**></comparacion>

Los caracteres < y & siempre se interpretan como especiales, excepto en secciones CDATA o mediante las *referencias a carácter* que hemos comentado en el apartado de codificación.

Los caracteres de comillas simples o dobles(**'** y **"**) no pueden aparecer en valores y atributos que usen como delimitador el mismo tipo de comillas. Para resolverlo utilizaremos como delimitador el carácter contrario al que necesitamos incluir en el texto. Si necesitamos los dos tipos de comillas tendremos que recurrir a las *referencias a carácter.*

Aunque en caso de conflicto se puede utilizar la referencia a carácter (&#código-decimal;), es recomendable usar las entidades predefinidas (llamadas también referencias a entidades) que representan a estos caracteres. Comienzan por & y terminan con ;

Son las siguientes:

Entidad	Carácter
&lt;	<
&gt;	>
&amp;	&
&quot;	“
&apos;	'

8. Comentarios

No forman parte del texto. Comienzan <!-- y terminan con -->

<!-- esto es un comentario -->

No pueden anidarse.

9. Documento bien formado

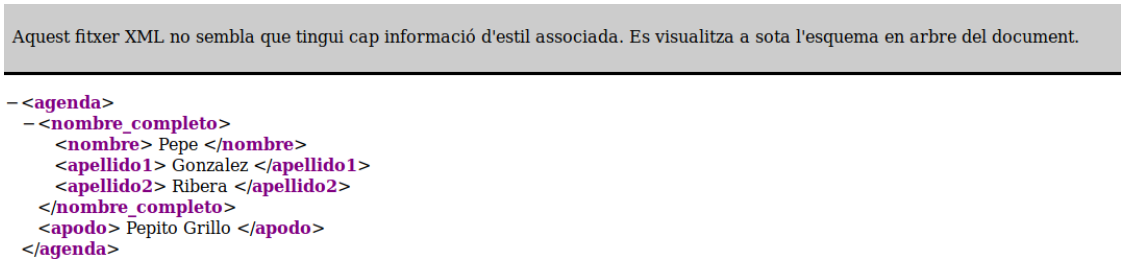
Nuestro objetivo va a ser crear documentos XML que sean ***bien formados*** y ***válidos***.

En el próximo tema veremos como validar documentos XML mediante DTD, pero de momento lo que podemos hacer es comprobar que sean ***bien formados***

A continuación se enumeran algunas reglas que deben cumplir un documento para que se le considere bien formado:

- Los elementos deben esta anidados adecuadamente.
- Los valores de los atributos deben encerrarse entre comillas simples o dobles.
- Todo elemento debe tener una etiqueta de fin o utilizar la etiqueta de elemento vacío.
- El elemento debe tener un único elemento raíz.
- Todo texto debe estar incluido en un elemento.

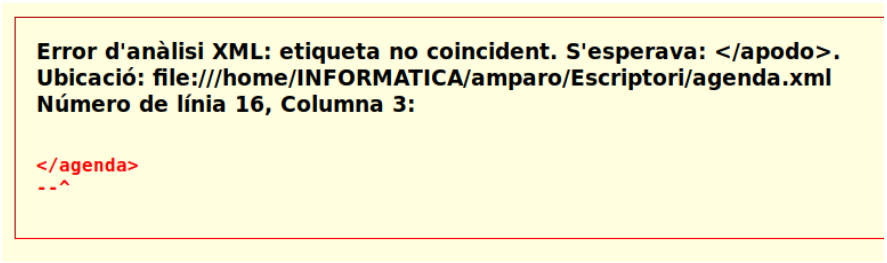
La comprobación de que un documento está bien formado la podemos hacer simplemente con el navegador. Si editamos el ejemplo 1 con un editor de texto plano (bloc de notas, notepad, gedit, etc.) sin la linea de DOCTYPE que es para llamar a la validación, como veremos en el tema siguiente, y luego lo abrimos con un navegador (Firefox) visualizamos el siguiente resultado:



Aquest fitxer XML no sembla que tingui cap informació d'estil associada. Es visualitza a sota l'esquema en arbre del document.

```
−<agenda>
  −<nombre_completo>
      <nombre> Pepe </nombre>
      <apellido1> Gonzalez </apellido1>
      <apellido2> Ribera </apellido2>
    </nombre_completo>
    <apodo> Pepito Grillo </apodo>
  </agenda>
```

Sin embargo, si hubiésemos cometido cualquier error, como cambiar el nombre de cierre de una etiqueta, obtenemos:



Error d'anàlisi XML: etiqueta no coincident. S'esperava: </apodo>.
Ubicació: file:///home/INFORMATICA/amparo/Escriptori/agenda.xml
Número de línia 16, Columna 3:

```
</agenda>
--^
```

Este tratamiento de visualización será distinto en cada navegador, incluso muchas veces cambia de una versión a otra del mismo navegador.