Sintaxis de XML

Estructura del documento XML

Un documento XML tiene siempre esta estructura básica:

```
Declaración / Versión XML
Nodo raíz
  Sub-elementos
  ...
Cierre del nodo raíz
```

Declaración XML

Define la versión XML y las características del documento:

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

Esta etiqueta tiene los siguientes atributos:

`version`

Obligatorio a no ser que este ya incluido en otro documento.

`encoding`

Opcional. Es la codificación de caracteres con la que se ha guardado el documento.

`standalone`

Opcional. Indica si un documento va acompañado de un DTD o no lo necesita (si no tiene DTD asociado, `standalone` valdrá `yes`)

Advertencia

A veces la codificación de caracteres del archivo y lo que dice la declaración XML no coinciden y a la hora de procesar el documento hay problemas al interpretar ciertas letras o símbolos. Si ocurre esto, conviene comprobar que realmente el documento tenga la codificación que dice.

Nodo raíz

Todos los elementos de un documento XML deben estar contenidos entre las etiquetas de apertura y cierre del nodo raíz, es decir, cuelgan del nodo padre que contiene a todos los demás como si fuera una estructura de árbol.

Sub-elementos hijo

Contendrán todos los elementos con sus datos y atributos.

Final del elemento raíz

Como todas las etiquetas deben tener su etiqueta de cierre, el nodo raíz también.

Elementos XML

Etiquetas de cierre

En XML es sintácticamente ilegal omitir la etiqueta de cierre, a diferencia de en HTML donde algunos elementos pueden no tener etiqueta de cierre. El siguiente ejemplo sería válido en HTML:

```
<p>Esto es un mensaje.
<p>Enviado de Pedro a Elisa.
```

Sin embargo en XML, todos los elementos deben tener etiqueta de cierre:

```
<p>Esto es un mensaje.</p>
<p>Enviado de Pedro a Elisa.</p>
```

En el caso de elementos vacíos, se admite una única etiqueta en lugar del par de etiquetas de apertura/cierre. En esos casos, la etiqueta debe escribirse como `<etiqueta />` (poniendo el carácter de etiqueta de cierre después del nombre de la etiqueta).

> **Nota**
>
> La declaración del documento XML no tiene etiqueta de cierre. No es un error, simplemente la declaración no forma parte del documento XML y no debe tener etiqueta de cierre.

Mayúsculas / minúsculas

A diferencia de HTML, XML distingue entre mayúsculas y minúsculas:

```
<Mensaje>Esto NO es correcto </mensaje>
<mensaje>Esto SI es correcto </mensaje>
```

Anidamiento de etiquetas

El anidamiento incorrecto de etiquetas no tiene sentido en XML. Aunque en HTML algunos elementos pueden anidarse de forma incorrecta:

```
<b><i>Este texto se ve en letra cursiva y negrita</b></i>
```

En XML todas las etiquetas deben anidarse correctamente:

```
<b><i>Este texto se ve en letra cursiva y negrita</i></b>
```

Nodo raíz

La primera etiqueta en un documento XML es la etiqueta raíz. Todos los documentos XML deben contener un par de etiquetas para definir el elemento raíz.

Elementos hijo

Estos deben tener la siguiente estructura:

```
<raiz>
  <hijo>
    <nieto> ... </nieto>
  </hijo>
</raiz>
```

Uso de comillas en XML

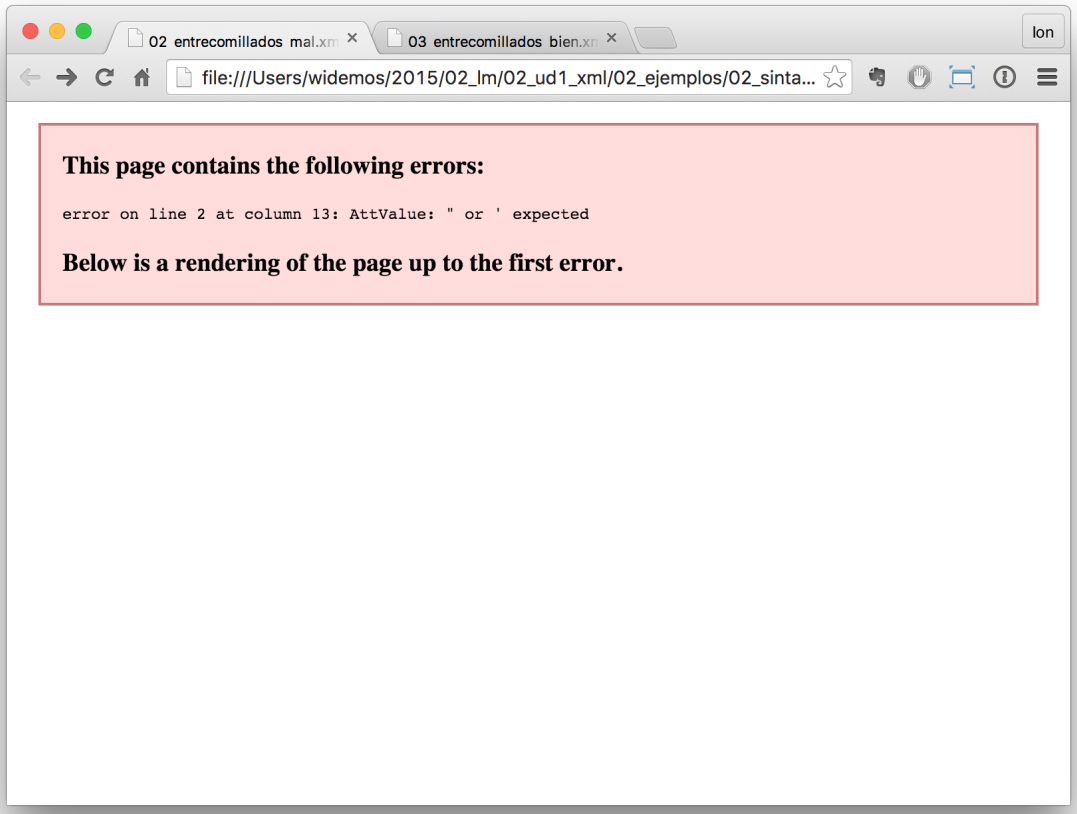
Las etiquetas XML pueden tener atributos con el formato:

```
<elemento atributo="valor"> ... </elemento>
```

En los ejemplos siguientes podemos ver la sintaxis correcta e incorrecta de un documento XML.

Este primer ejemplo es incorrecto porque los valores de los atributos no están entrecomillados:

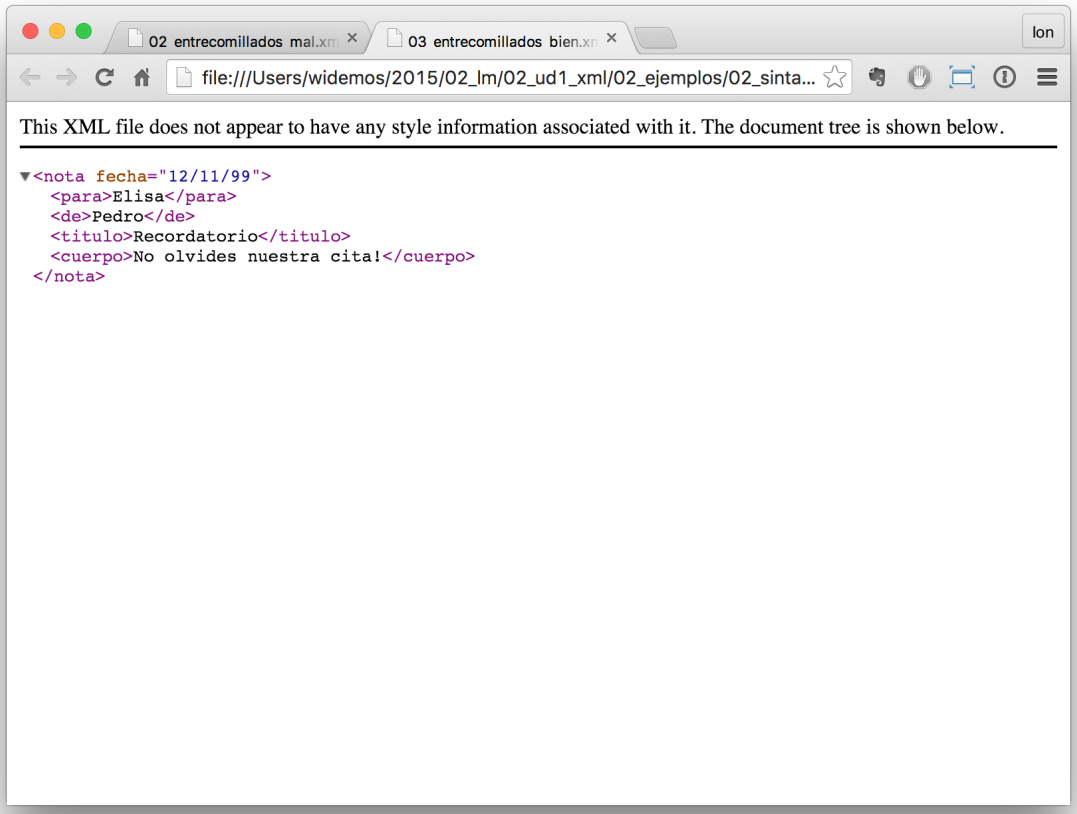
```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<nota fecha=12/11/99>
  <para>Elisa</para>
  <de>Pedro</de>
  <titulo>Recordatorio</titulo>
  <cuerpo>No olvides nuestra cita!</cuerpo>
</nota>
```



XML con entrecomillado incorrecto.

Aquí vemos el mismo ejemplo pero con una sintaxis correcta. Los atributos de la etiqueta `<nota>` están delimitados por comillas:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<nota fecha="12/11/99">
  <para>Elisa</para>
  <de>Pedro</de>
  <titulo>Recordatorio</titulo>
  <cuerpo>No olvides nuestra cita!</cuerpo>
</nota>
```



XML con entrecomillado correcto.

Conservación de espacios

En XML los espacios en blanco se conservan, no son truncados a un espacio único a diferencia de HTML, donde los espacios en blanco seguidos, así como caracteres de tabulación y saltos de línea, son comprimidos a un único espacio en blanco.

Formato de ficheros XML

Son ficheros de texto plano, lo que permite trabajar con ellos desde cualquier editor de texto.

Elementos extensibles

Los documentos XML pueden ampliarse para incluir más información. Vamos a estudiar el ejemplo previo de la nota enviada de Pedro a Elisa:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<nota>
  <para>Elisa</para>
  <de>Pedro</de>
  <titulo>Recordatorio</titulo>
  <cuerpo>No olvides nuestra cita!</cuerpo>
</nota>
```

Imaginemos que hemos creado una aplicación que extrae los elementos `<para>`, `<de>` y `<cuerpo>`. Supongamos que el autor añade una información extra, `<fecha>`:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<nota>
  <fecha>27 de mayo del 2010</fecha>
  <para>Elisa</para>
  <de>Pedro</de>
  <titulo>Recordatorio</titulo>
  <cuerpo>No olvides nuestra cita!</cuerpo>
</nota>
```

La aplicación no tiene que fallar ya que debería poder localizar los elementos `<para>`, `<de>` y `<cuerpo>` en el documento y producir la misma salida.

Relación semántica entre elementos

Los elementos tienen entre sí relaciones del tipo padre-hijo. Para entender la terminología XML es importante conocer las relaciones entre los diferentes elementos de un documento, como se identifican y como son descritos los elementos de contenido (datos).

Contenido de los elementos

Un elemento puede contener:

- Nada (elemento vacío).
- Datos.
- Subelementos XML.
- Atributos.

No tiene porque incluir sólo una de estas clases, puede haber varias mezcladas.

En el ejemplo siguiente, el elemento `<libro>` contiene dos elementos: `<producto>` y `<capitulo>`. El elemento `<producto>` es un elemento vacío, porque no contiene ningún dato. En este caso, tiene los atributos `id` y `medio`, cada uno de ellos con sus valores entrecomillados.

El documento XML que describe el libro sería:

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<libro>
  <titulo>El mundo de XML</titulo>
  <producto id="33-657" medio="papel"></producto>
  <capitulo>Introduccion a XML
    <par>Que es html</par>
    <par>Que es xml</par>
  </capitulo>
</libro>
```

Reglas de nombrado de elementos

Los elementos XML deben seguir las siguientes reglas de nombrado:

- Los nombres pueden contener letras, números y otros caracteres.
- Los nombres no pueden comenzar con un número, con el carácter `_` (guión bajo) o con los caracteres `xml` (ni variaciones tipo `XML`, `Xml`...)
- Los nombres no pueden contener espacios (se utiliza el guíon bajo `_` para separar palabras).

A la hora de nombrar los elementos es importante seguir algunos consejos sencillos, que pueden facilitar las cosas:

- Puede utilizarse cualquier nombre, no hay palabras reservadas, pero conviene utilizar nombres descriptivos para facilitar la comprensión de los datos.
- Puede ayudar el utilizar el guión bajo para separar nombres de varias palabras (`primer_apellido`, `segundo_apellido`, ...).
- Evitar el uso de los caracteres `-` y `.` dado que el software de tratamiento de los datos lo puede identificar como símbolos aritméticos o como propiedades de objetos.
- Los nombres de los elementos pueden ser tan largos como se desee, pero no es conveniente exagerar. Es mejor que sean cortos y simples (si no hay ambigüedad, no conviene usar nombres como `el_titulo_del_libro` cuando se puede utilizar `titulo`).
- Los caracteres no pertenecientes al alfabeto latino, son perfectamente válidos (ñ, á, ô, etc.) Sin embargo conviene asegurarse de que el software de tratamiento de los datos no tenga problemas con dichos caracteres.
- El carácter `:` no debería utilizarse en la denominación de los elementos, dado que está reservado para los *namespaces*.

Atributos XML

En HTML es habitual que las etiquetas tengan atributos que proporcionan información adicional sobre la propia etiqueta.

Por ejemplo en la etiqueta,

```
<IMG SRC="mi_casa.gif">
```

el atributo `src` proporciona información adicional sobre la imagen. En este caso nos dice el fichero que la contiene.

De la misma forma, los atributos en etiquetas XML proporcionan información sobre la propia etiqueta que los contiene:

```
<img src="computer.gif">
<a href="demo.asp">
```

Los atributos aportan información que no es parte de los datos:

```
<fichero tipo="gif">mi_casa.gif</fichero>
```

En el caso anterior, el tipo de fichero de imagen no es importante para los datos, pero sí lo es para el software que manipula la información.

Tipos de entrecomillado

Ya se ha comentado anteriormente que todos los valores de los atributos deben estar entrecomillados. Pero el tipo de comillas utilizado es irrelevante; podemos utilizar tanto comillas simples como comillas dobles pero, eso sí, debemos utilizar el mismo tipo de comillas en ambas

partes de la expresión entrecomillada.

Estos formatos serían admitidos:

```
<fichero tipo="gif">mi_casa.gif</fichero>
<fichero tipo='gif'>mi_casa.gif</fichero>
```

Pero no estos:

```
<fichero tipo="gif'>mi_casa.gif</fichero>
<fichero tipo='gif">mi_casa.gif</fichero>
```

Las dobles comillas suelen ser más utilizadas, pero en ocasiones es necesario utilizar comillas sencillas, como en el ejemplo siguiente:

```
<gangster nombre='Miguel "Pistolas" Fernandez'>
```

¿Elementos o atributos?

Veamos algunos objetos:

```
<persona sexo="femenino">
    <nombre>Elisa</nombre>
    <apellido>Lopez</apellido>
</persona>
```

```
<persona>
    <sexo>femenino</sexo>
    <nombre>Elisa</nombre>
    <apellido>Lopez</apellido>
</persona>
```

En el primer ejemplo, el sexo es un atributo del elemento persona. En el segundo, sexo es un elemento hijo del elemento persona. No existen reglas sobre cuando utilizar atributos o elementos hijos. Sin embargo, como norma general, se debería tender a utilizar los elementos hijos en lugar de los atributos.

Además, el uso de atributos tiene algunos problemas:

- Los atributos no pueden contener generalmente valores múltiples, mientras que los elementos sí.
- Los atributos son difíciles de expandir en el caso de que se deeen hacer cambios futuros en la estructura de los datos.
- Los atributos no permiten estructurar la información.
- Los atributos son más difíciles de manipular por las aplicaciones.
- Los valores de los atributos son difíciles de verificar frente a una DTD.

Sin embargo, hay ocasiones en las que el uso de atributos si puede ser recomendable. Veamos el siguiente ejemplo para entenderlo:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<mensajes>
  <nota ID="001">
    <para>Elisa</para>
    <de>Pedro</de>
    <titulo>Recordatorio</titulo>
    <cuerpo>No olvides nuestra cita!</cuerpo>
  </nota>
  <nota ID="002">
    <para>Juan</para>
    <de>Francisco</de>
    <titulo>Cita</titulo>
    <cuerpo>Quedamos a comer en el Restaurante de abajo.</cuerpo>
  </nota>
</mensajes>
```

El atributo `ID` en este ejemplo es solamente un contador de mensajes y no una parte de los datos. En este caso sí podemos decir que el uso de los atributos está recomendado. La información que contiene es los que se denomina *metainformación* (información sobre la información).

Comentarios

Para poder documentar un programa XML que sirva de guía para comprenderlo, pondríamos las siguientes etiquetas:

```
<!-- COMENTARIOS -->
```

Donde pone `COMENTARIOS` añadimos todo nuestro texto. Evitar utilizar guiones en los comentarios para evitar conflictos.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<email>
  <!--  Destinatario del mensaje  -->
  <para>Elisa</para>

  <!--  Remitente del mensaje  -->
  <de>Pedro</de>

  <titulo>Recordatorio</titulo>
  <cuerpo>No olvides nuestra cita</cuerpo>
</email>
```

Caracteres especiales de XML

Hay una serie de caracteres que XML no reconoce y los considera como ilegales. Para poder incluírlos, se utilizan una serie de referencias.

Si por ejemplo introducimos un símbolo de menor `<` dentro de una etiqueta el *parser* dará como respuesta un mensaje de error porque considera que si hay un símbolo de menor, es el comienzo de una nueva etiqueta.

Por ejemplo algo que produciría un error es:

```
<mensaje>si salario <1000 entonces </mensaje>
```

Para solucionar esto sustituimos dicho símbolo por una referencia:

```
<mensaje>si salario &lt;1000 entonces </mensaje>
```

Hay 5 referencias predeterminadas:

Caracter	Referencia	Unicode
<	<	<
>	>	>
&	&	&
'	'	'
"	"	"

Para más información se puede consultar:

http://www.w3schools.com/charsets/ref_utf_basic_latin.asp

Ejemplo completo de documento XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<!-- LISTADO DE PERSONAL AUTORIZADO -->
<personal>
  <persona id="01">
    <nombre>&quot; Directora &quot; Nerea</nombre>
    <apellido>Urbieta</apellido>
    <direccion>Gran Via 5, Bilbo</direccion>
    <matricula>0 &#8364;</matricula>
  </persona>
  <persona id="100">
    <nombre>Idoia</nombre>
    <apellido>Elorza</apellido>
    <direccion>Getaria Kalea, Donostia</direccion>
    <matricula>800 &#8364;</matricula>
  </persona>
  <persona id="101">
    <nombre>Nagore</nombre>
    <apellido>Dorronsoro</apellido>
    <direccion>Dato Kalea 6, Gasteiz</direccion>
    <matricula>800 &#8364;</matricula>
  </persona>
  <persona id="102">
    <nombre>Eli</nombre>
    <apellido>Agirre</apellido>
    <direccion>Dato Kalea 8, Gasteiz</direccion>
    <matricula>800 &#8364;</matricula>
  </persona>
</personal>
```