

# **XML**

## **eXtensible Markup Language**

# ¿Qué es XML?

**Es un lenguaje desarrollado y lanzado en 1998 por la W3C (World Wide Web Consortium) basado en SGML (Standard Generalized Markup Language).**

**Se utiliza para el almacenamiento e intercambio de datos estructurados entre distintas plataformas.**

**XML es un metalenguaje (lenguaje para definir otros lenguajes). Los lenguajes creados son llamados dialectos XML, ejemplos: GML (Geography ML), MathML (Mathematical ML), SVG (Scalable Vector Graphics), XHTML (eXtensible HyperText ML) ...**

# ELEMENTOS

**Los documentos XML están formados por texto plano y contienen marcas o etiquetas definidas por el programador. Las marcas deben de ser descriptivas y se escriben entre los símbolos < , > y /. Además de la marca está el valor y marca y valor constituyen el elemento**

**Ejemplo: <nombre>Manuel</nombre> [Manuel es el valor que se almacena en la etiqueta nombre (Apertura y cierre).**

**Si un elemento no contiene ningún valor se dice que está vacío.**

**Ejemplo: <nombre></nombre>**

# Relaciones entre los elementos (padre-hijo)

**<persona>**

**Los datos son:**

**<nombre>Paco</nombre>**

**<mujer />**

**<fecha-Nacimiento>**

**<día>25</día>**

**<mes>4</mes>**

**<año>1996</año>**

**</fecha-Nacimiento>**

**<ciudad>Logroño</ciudad>**

**</persona>**

**El elemento “persona” tiene 4 hijos (nombre,mujer, fecha-Nacimiento y ciudad).**

**El elemento “mujer” está vacío.**

**El elemento fecha-Nacimiento tiene 3 hijos (día, mes y año) que a su vez serán nietos del elemento “persona”.**

**Todo documento XML tiene un elemento raíz del que descienden todos los demás, en este ejemplo es “persona”.**

**Se dice que el elemento “persona” tiene contenido mixto al contener otros elementos y texto (Los datos son:)**

# Reglas sintácticas

**En un documento XML, todos los nombre de los elementos son «case sensitive»**

**Los nombres de los elementos pueden contener: letras minúsculas y mayúsculas, números, puntos, guiones medios y bajos, los dos puntos : (aunque su uso queda reservado para definir espacios de nombre).**

**El primer carácter tiene que ser una letra o un gión bajo.**

INCORRECTO

```
<Ciudad>Pamplona</ciudad>
<dia>18</dia>
<mes>6<mes/>
<ciudad>Pamplona</finciudad>
<_rojo>
<2colores>Rojo y Naranja</2colores>
<_Aficiones >Cine, Bailar, Nadar</_Aficiones >
<persona><nombre>Elsa</persona></nombre>
<color favorito>azul</color favorito>
```

CORRECTO

```
<Ciudad>Pamplona</Ciudad>
<dia>18</dia>
<mes>6</mes>
<ciudad>Pamplona</ciudad>
<_rojo/>
<colores2>Rojo y Naranja</colores2>
<Aficiones >Cine, Bailar, Nadar</Aficiones >
<persona><nombre>Elsa</nombre></persona>
<color.favorito>azul</color.favorito>
<color-favorito>azul</color-favorito>
<color_favorito>azul</color_favorito>
```

# Reglas sintácticas. Recomendaciones

**Aunque estén permitidas las palabras con tilde se recomienda no utilizarlas para si se prevé que se pueden utilizar con aplicaciones sajonas.**

**El guion medio se podría confundir con el signo menos.**

**El punto (.) se podría interpretar como una propiedad.**

**Ejemplo: color.favorito se podría interpretar como que favorito es una propiedad del objeto color.**

# Atributos

**Los elementos de un documento XML pueden tener atributos definidos en la etiqueta de inicio. Un atributo proporciona información extra sobre el elemento que lo contiene. La sintaxis de un atributo es la misma que la de los elementos.**

**Cada atributo, en un elemento, debe ser único.**

```
<producto codigo="G45">  
  <nombre color="negro" precio="12.56">Gorro de lana</nombre>  
</producto>
```

# Declaración XML

**Se escribe al principio del documento. No es una instrucción de procesamiento.**

**<?xml versión="1.0" encoding="UTF-8"?>**

```
<?xml version="1.0" encoding="UTF-8"?>
<persona profesion="cantante">
  <nombre>Elsa</nombre>
  <mujer/>
  <fecha_de_nacimiento>
    <dia>18</dia>
    <mes>6</mes>
    <año>1996</año>
  </fecha_de_nacimiento>
  <ciudad>Pamplona</ciudad>
</persona>
```

**UTF: Unicode Transformation Format**



# Declaración XML. Atributo standalone

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

**La declaración no es obligatoria pero si se pone el atributo version es obligatorio**

**El atributo standalone puede ser “yes” o “no”.**

**Si es “yes” significa que el documento es independiente de otros, como DTD o schemas.**

**encoding y standalone no son obligatorios. Los valores por omisión son UTF-8 y “no”, respectivamente. Si se ponen, standalone se pondrá el último.**

# Instrucciones de Procesamiento

**Una instrucción de procesamiento (Processing Instruction) sirve para indicar cierta información al programa que ejecute el documento XML.**

**Las instrucciones de proceso se escriben entre el caracter de apertura <? y el de cierre ?>**

**Por ejemplo se podría usar para asociar un archivo CSS a un documento XML**

**<?xml-stylesheet type="text/css" href="estilo-animales.css"?>**

**Archivo: estilo-animales.css**

```
nombre{color:blue;font-size:40px}  
patas{color:red;font-size:22px}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/css" href="estilo-animales.css"?>  
<animales>  
  <animal>  
    <nombre>perro</nombre>  
    <patas>4</patas>  
  </animal>  
  <animal>  
    <nombre>pato</nombre>  
    <patas>2</patas>  
  </animal>  
  <animal>  
    <nombre>ballena</nombre>  
    <patas>0</patas>  
  </animal>  
</animales>
```

**Archivo: animales.xml**

# Referencias a Entidades

**En XML existen caracteres que son especiales por su significado y, para escribirlos se pueden utilizar las referencias a las entidades siguientes:**

Referencias a entidades en XML		
<i>Carácter</i>	<i>Entidad</i>	<i>Referencia a entidad</i>
<i>&lt; (menor que)</i>	<i>lt (less than)</i>	<i>&amp;lt;</i>
<i>&gt; (mayor que)</i>	<i>gt (greater than)</i>	<i>&amp;gt;</i>
<i>" (comilla doble)</i>	<i>quot (quotation mark)</i>	<i>&amp;quot;</i>
<i>' (comilla simple)</i>	<i>apos (apostrophe)</i>	<i>&amp;apos;</i>
<i>&amp; (ampersand)</i>	<i>amp (ampersand)</i>	<i>&amp;amp;</i>

# Referencias a Entidades (Recomendaciones)

**1.- Se recomienda &lt; en vez de <**

**2.- Se recomienda &gt; en vez de >**

**3.- Usar &amp; en vez de &**

**4.- Entre comillas dobles no se puede utilizar las comillas doble:**

**<dato carácter="mis comillas (")" />**

**Hay que poner <dato carácter="mis comillas (&quot;)" />**

**Lo mismo ocurre con las comillas simples, sustituyendo &quot; por &apos;**

**5.- Si se puede poner comillas simples entre dobles o dobles entre simples:**

**<dato carácter="mis comillas (')" /> o <dato carácter='mis comillas (")' />**

# Referencias de caracteres

**En un documento XML se pueden escribir referencias de caracteres Unicode con los símbolos &#, seguidos del valor decimal o hexadecimal del carácter Unicode que se quiera representar y añadiendo un ; al final.**

```
<?xml version="1.0" encoding="UTF-8"?>
<productos>
  <nombre precio="12.56&#8364;">Gorro de lana</nombre>
  <nombre precio="16.99&#x20AC;">Gorro polar</nombre>
</productos>
```

```
<productos>
  <nombre precio="12.56€">Gorro de lana</nombre>
  <nombre precio="16.99€">Gorro polar</nombre>
</productos>
```

# COMENTARIOS

Para incluir comentarios: **<! -- Mis comentarios -- >**

En los comentarios **no se permite** incluir dos guiones seguidos:

**<! -- Mis -- comentarios -- >**

**Tampoco se permiten** los comentarios dentro de las etiquetas de un elemento:

**<mujer <!-- elemento booleano -- > />**

# Sección CDATA

**Un documento XML puede contener secciones CDATA (Character DATA) para escribir texto que no se desea que sea analizado. No se pueden anidar CDATA ni incluir los caracteres ]]>**

**Sintaxis: <![CDATA[“texto a incluir”]]>**

**Ejemplo:**

**Documento XML**

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo_CDATA>
<![CDATA[
#include <stdio.h>
int main()
{
    float nota;
    printf( "\n    Introduzca nota (real): " );
    scanf( "%f", &nota );
    if ( 5 <= nota )
        printf( "\n    APROBADO" );
    return 0;
}
]]>
</ejemplo_CDATA>
```

**Resultado en navegador**

```
▼<ejemplo_CDATA>
  ▼<![CDATA[
    #include <stdio.h> int main() { float nota;
    printf( "\n Introduzca nota (real): " ); scanf(
    "%f", &nota ); if ( 5 <= nota ) printf( "\n
    APROBADO" ); return 0; }
  ]]>
</ejemplo_CDATA>
```

# Espacios de nombres

**Los documentos XML se pueden combinar entre sí, pudiendo coincidir el nombre de algunos elementos.**

**Para evitar el conflicto de nombres se pueden utilizar los espacios de nombres (XML Namespaces).**

**La sintaxis:**

**xmlns:prefijo="URI"**



# Espacios de nombres

```
<?xml version="1.0" encoding="UTF-8"?>
<e1:ejemplo xmlns:e1="http://www.abrirllave.com/ejemplo1"
  xmlns:e2="http://www.abrirllave.com/ejemplo2">

  <e1:carta>
    <e1:palo>Corazones</e1:palo>
    <e1:numero>7</e1:numero>
  </e1:carta>

  <e2:carta>
    <e2:carnes>
      <e2:filete_de_tenera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</e1:ejemplo>
```

**Hay que observar que xmlns es un atributo de la etiqueta de inicio correspondiente al elemento <ejemplo> y se han definido dos espacios de nombre que hacen referencia a los siguientes URI:**

**<http://www.abrirllave.com/ejemplo1>**

**<http://www.abrirllave.com/ejemplo2>**

**Los prefijos son e1 y e2. Estos prefijos se añaden en el documento xml <e1:carta>, <e2:pescados> ...**

**Los URI no tienen porqué ser direcciones reales de internet, su función es ser únicos.**

# Espacios de nombres en elementos distintos del raíz

**Los espacios de nombre se pueden definir en el elemento raíz o, directamente, en los elementos que los vayan a utilizar.**

**Se pueden crear tantos espacios de nombre como sean necesarios.**

```
<?xml version="1.0" encoding="UTF-8"?>
<el:ejemplo xmlns:el="http://www.abrirllave.com/ejemplo1">

  <el:carta>
    <el:palo>Corazones</el:palo>
    <el:numero>7</el:numero>
  </el:carta>

  <e2:carta xmlns:e2="http://www.abrirllave.com/ejemplo2">
    <e2:carnes>
      <e2:filete_de_tenera precio="12.95"/>
      <e2:solomillo_a_la_pimienta precio="13.60"/>
    </e2:carnes>
    <e2:pescados>
      <e2:lenguado_al_horno precio="16.20"/>
      <e2:merluza_en_salsa_verde precio="15.85"/>
    </e2:pescados>
  </e2:carta>

</el:ejemplo>
```

# Espacios de nombres por Defecto

**Se puede definir un espacio de nombres por defecto:**

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

</ejemplo>
```

**El espacio de nombres afecta al elemento donde se define y a todos sus sucesores.**

# Varios Espacios de nombres por Defecto

Se pueden definir más de un espacio de nombres por defecto:

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

  <carta xmlns="http://www.abrirllave.com/ejemplo2">
    <carnes>
      <filete_de_tenera precio="12.95"/>
      <solomillo_a_la_pimienta precio="13.60"/>
    </carnes>
    <pescados>
      <lenguado_al_horno precio="16.20"/>
      <merluza_en_salsa_verde precio="15.85"/>
    </pescados>
  </carta>
</ejemplo>
```

# Espacios de nombres. Elementos no afectados

Cuando se quiera indicar que un elemento no está afectado por un espacio de nombres, se indicará: **xmlns=""**

```
<?xml version="1.0" encoding="UTF-8"?>
<ejemplo xmlns="http://www.abrirllave.com/ejemplo1">

  <carta>
    <palo>Corazones</palo>
    <numero>7</numero>
  </carta>

  <carta xmlns="http://www.abrirllave.com/ejemplo2">
    <carnes>
      <filete_de_tenera precio="12.95"/>
      <solomillo_a_la_pimienta precio="13.60"/>
    </carnes>
    <pescados xmlns="">
      <lenguado_al_horno precio="16.20"/>
      <merluza_en_salsa_verde precio="15.85"/>
    </pescados>
  </carta>

</ejemplo>
```

**El elemento <pescados> y sus hijos no están afectados por ningún espacio de nombre.**

# Espacios en blanco

En un documento XML, los espacios en blanco, las tabulaciones y los retornos de carro pueden ser tratados de un modo especial.

Si se dejan espacios en blanco o se insertan tabulaciones en los valores de los elementos o entre elementos o en los valores de un atributo, los navegadores lo interpretarán como un solo espacio.

Si se quieren mantener los espacios se usará el atributo **xml:space**

El atributo **xml:space** puede tener dos valores: **preserve** y **default**.

**Default:** Deja a la aplicación que trata el XML que actúe con los espacios según su criterio.

**Preserve:** XML indica que se tienen que respetar los espacios existentes.

```
<clasificacion xml:space="preserve">
1      Fernando Alonso      1:55.341
2      Lewis Hamilton       1:55.729
3      Sebastian Vettel     1:56.122
</clasificacion>
```



# Documentos XML bien formados

**Un documento XML está bien formado (well-formed document) cuando no tiene errores de sintaxis, es decir:**

- Los nombres de los elementos y sus atributos deben estar escritos correctamente.
- Los valores de los atributos deben estar escritos entre comillas dobles o simples.
- Los atributos de un elemento deben separarse con espacios en blanco.
- Se tienen que utilizar referencias a entidades donde sea necesario.
- Tiene que existir un único elemento raíz.
- Todo elemento debe tener un elemento padre, excepto el elemento raíz.
- Todos los elementos deben tener una etiqueta de apertura y otra de cierre.
- Las etiquetas deben estar correctamente anidadas.
- Las instrucciones de proceso deben estar escritas de forma correcta.
- La declaración XML debe estar en la primera línea escrita correctamente.
- Las secciones CDATA y los comentarios deben estar correctamente escritos.

# Documentos XML válidos

Un documento XML es válido (valid document) cuando, además de **estar bien formado, no incumple ninguna de las normas establecidas en su estructura.**

**La estructura se define utilizando:**

**DTD (Document Type Definition)**

**XML Schema**



# Ejemplos de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<marcadores>
  <pagina>
    <nombre>Abrirllave</nombre>
    <descripcion>Tutoriales de informática.</descripcion>
    <url>http://www.abrirllave.com/</url>
  </pagina>
  <pagina>
    <nombre>Wikipedia</nombre>
    <descripcion>La enciclopedia libre.</descripcion>
    <url>http://www.wikipedia.org/</url>
  </pagina>
  <pagina>
    <nombre>W3C</nombre>
    <descripcion>World Wide Web Consortium.</descripcion>
    <url>http://www.w3.org/</url>
  </pagina>
</marcadores>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<equipos_de_futbol>
  <equipo>
    <nombre>Levante</nombre>
    <ciudad>Valencia</ciudad>
    <entrenador></entrenador>
    <jugadores>
      <jugador posicion="portero">
        <nombre>Javier Jiménez</nombre>
        <nacionalidad>española</nacionalidad>
      </jugador>
      <jugador posicion="delantero">
        <nombre>Nong</nombre>
        <nacionalidad>camerunés</nacionalidad>
      </jugador>
    </jugadores>
  </equipo>
</equipos_de_futbol>
```