



UD1.2 XML

LMSXI

IES de Teis

Vigo

Curso 2021-2022

XML: Características

- eXtensible Markup Language
- Metalenguaje de marcas
- Subconjunto de SGML
- Ejemplo:

```
1<editorial>
2  <libro>
3      <titulo>La vida está en otra parte</titulo>
4      <autor>Milan Kundera</autor>
5      <fechaPublicacion anho="1973" mes="2" dia="12" />
6  </libro>
7  <libro>
8      <titulo>Cancionero popular</titulo>
9      <autor fechaNacimiento=""></autor>
10     <fechaPublicacion anho="1980" />
11 </libro>
12 <libro>
13     <titulo>Mis Memorias</titulo>
14     <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
15     <fechaPublicacion anho="" />
16     <!-- aún por publicar -->
17 </libro>
18 </editorial>
```

XML: versiones

- El Consorcio World Wide Web (W3C) es una comunidad internacional que desarrolla estándares abiertos para asegurar el crecimiento a largo plazo de la Web
- En concreto de XML, existen 2 versiones:
 - Versión 1.0: inicialmente del 1998, última del 2008
 - Versión 1.1: menos utilizada, con soporte para permitir mayor número de caracteres (en codificaciones UNICODE 4.0).

XML: Elementos

- Un elemento está formado **normalmente** por 2 etiquetas, una de apertura y otra de cierre.
 - La etiqueta de apertura va entre los símbolos menor que ‘<’ y mayor que ‘>’
 - La etiqueta de cierre va entre los símbolos menor que seguido de una barra ‘</’ y mayor que ‘>’
 - Por ejemplo: titulo sera aquí un elemento

```
<titulo>La vida está en otra parte</titulo>
```
- Un documento XML ha de tener obligatoriamente un único **elemento raíz**. (*editorial* en el documento XML de ejemplo)



XML: Elementos

- Un elemento puede tener independiente o simultáneamente:
 - Texto entre la etiqueta de apertura y cierre
 - Atributos
 - Otros elementos anidados

XML: Nombres de elementos

- Los nombres de elementos deben seguir algunas reglas:
 - La etiqueta de apertura y cierre deben tener el mismo nombre
 - Mayúsculas y minúsculas se consideran caracteres diferentes. Se dice que XML es *case-sensitive*
 - Deben comenzar por un carácter **alfabético** (acentuado o no) o guion bajo ‘_’ **(no pueden comenzar por un número)**
 - No pueden comenzar por XML, xML, Xml,... o cualquier combinación de esos 3 caracteres
 - Pueden contener cualquier carácter alfanumérico (acentuado o no) y los caracteres ‘.’, ‘_’ y ‘-’
 - Aunque también pueden contener ‘:’, no se aconseja porque se usa para los prefijos de espacios de nombres
 - **RECOMENDACIÓN:** No utilizar acentos en los nombres de las etiquetas para evitar problemas de codificación

XML: Elementos vacíos

- Si no hubiese texto, se tratará de elementos vacíos. Son posibles dos notaciones:

- `<autor></autor>`

- `<autor/>`

XML: Atributos

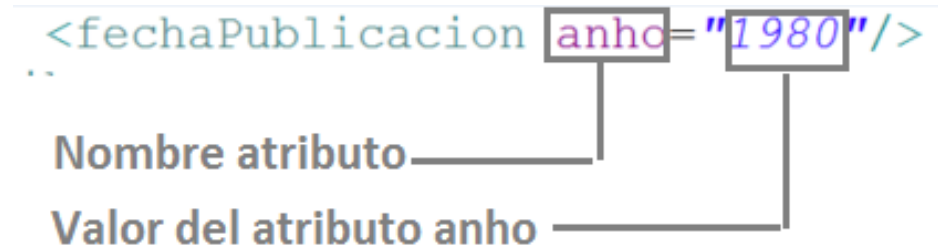
- Los atributos añaden información o características sobre los elementos. Por ejemplo, el atributo *año* especifica el año de la fecha de publicación

```
<fechaPublicacion año="1980"/>
```

- Los elementos pueden tener atributos o no.

XML: Atributos

- Se separan por al menos un espacio del nombre del elemento y entre atributos
- El valor del atributo debe seguir al nombre del atributo seguido a su vez de un =
- Su valor debe ir entre comillas dobles “ o simples ‘. Si se abre con un tipo, ha de cerrarse con el mismo tipo



The diagram shows an XML attribute value: `<fechaPublicacion año="1980"/>`. The attribute name `año` is highlighted with a black box, and the attribute value `"1980"` is also highlighted with a black box. Below the code, there are two labels with lines pointing to the corresponding parts of the attribute: `Nombre atributo` points to `año`, and `Valor del atributo año` points to `"1980"`.

- El nombre del atributo tiene las mismas restricciones que el nombre de los elementos y, además, no puede contener “ ni ‘.

XML: Atributos

- Debe existir siempre un valor entre comillas, permitiendo también la cadena vacía

```
<fechaPublicacion anho=""/>
```

- Puede haber más de un atributo en un mismo elemento
No importa el orden de los atributos dentro del mismo elemento

```
<fechaPublicacion anho="1973" mes="2" dia="12" />
```

- No puede haber más de un atributo con el mismo nombre para un mismo elemento

XML: ¿Elementos o atributos?

- Cuando se diseña un documento XML, podemos tener dudas sobre si utilizar un elemento o un atributo
 - Si la información sobre la que tenemos dudas, se puede descomponer en más elementos, se recomienda el uso de un elemento
 - Si es información periférica a la información principal, por ejemplo: un identificador de una base de datos (BD) o existe un n.º reducido de valores que puede tomar el valor del atributo (*true/false*) o si tiene un formato fijo (YYYY-MM-DD) => podría usarse un atributo
 - Los procesadores de XML permiten un mayor control sobre los elementos. Si existen dudas, se recomienda elemento.

XML: ¿Elementos o atributos?

- Por ejemplo, si tenemos que especificar tamaños de página reglados por la norma DIN A0, A2, etc. podríamos usar un atributo *size*:

```
<page size="a4">
```

- Pero tampoco sería incorrecto utilizar un elemento:

```
<page>  
  <size>a4</size>  
</page>
```

XML: ¿Elementos o atributos?

- Sin embargo, si se usase un tamaño arbitrario en diferentes unidades (mm, cm, etc), habría que descomponer más la información. Por ejemplo:

```
<page>  
    <size unit="cm">  
        <width>15</width>  
        <height>25</height>  
    </size>  
</page>
```

XML: ¿Elementos o atributos?

- Para estructurar la información de un correo electrónico que se ha guardado en una base de datos y desea exportarse en un documento XML. El identificador de la BD no es parte central del correo electrónico sino que lo podríamos considerar como un dato periférico y añadirlo como atributo

```
<mail id="0352">  
  <to>You</to>  
  <from>Me</from>  
  <heading>My message</heading>  
  <body>Dear friend, ...</body>  
</mail>
```

- También podría añadirse como un elemento anidado dentro del elemento mail



XML: Actividad

→ Actividad UD1.2.1-Agenda

XML: Texto comprendido entre elementos

- Se permite cualquier texto salvo “<” y “&”
- Deberían cambiarse por:
 - `&#valorUnicodeDec;`
 - `&#xvalorUnicodeHex;`
 - Entidades prefijadas:

Entidad	Carácter
<code>&amp;</code>	<code>&</code>
<code>&lt;</code>	<code><</code>
<code>&gt;</code>	<code>></code>
<code>&apos;</code>	<code>'</code>
<code>&quot;</code>	<code>"</code>

XML: Texto comprendido entre elementos

- Por ejemplo, para incluir el símbolo & en el texto:
 - <https://unicode.org/charts/> => Basic latin (ASCII)
 - & => 0026₍₁₆₎ (en hexadecimal)
 - Convertir a decimal=> 38₍₁₀₎
 - Conversor online
 - Calculadora científica
 - Podríamos usar cualquiera de estas 3 opciones:
`<nombre>Johnson & & & Johnson </nombre>`
 - Si lo abrimos con un navegador, deberíamos ver el símbolo & por cada una de las representaciones:
`<nombre>Johnson & & Johnson </nombre>`



XML: Texto comprendido en el valor de los atributos

- Se aplican las mismas consideraciones que para elementos
- Además, el valor de un atributo no puede contener el mismo tipo de comillas utilizado para delimitar el valor del mismo

XML: Comentarios

- Para añadir información adicional, que será ignorada por los procesadores de XML, pero útil para las personas que los lean, se puede añadir comentarios.
- Empiezan por "<!--" y terminan con "-->".

```
<!-- Esto es un comentario -->
```

XML: Instrucciones de procesamiento

- Parar información a las aplicaciones que procesen los documentos XML
- Pueden situarse en cualquier parte del documento, pero suelen colocarse al comienzo del mismo
- Empiezan por "<?" y terminan con "?>"

```
<?xml-stylesheet type="text/xsl" href="tarea05.xsl"?>
```

XML: Documentos bien formados

- Un documento XML bien formado o *well-formed* debe seguir las normas definidas en la especificación de XML del W3C para su versión correspondiente (XML 1.0 y XML 1.1)
- Algunas de ellas las hemos visto ya y otras las veremos a continuación
 - Debe haber un único elemento raíz, para simular una estructura de árbol.
 - Contiene a todos los demás elementos
 - Es posible encontrarlo referenciado como *el ejemplar*. Por ejemplo, el ejemplo de documento es *page*:

```
<page>  
  <size>a4</size>  
</page>
```

XML: Documentos bien formados

- Todos los elementos deben cerrarse de forma estructurada, es decir, en orden inverso a su apertura: el último elemento en abrir será el primero en cerrar
- Por ejemplo: No sería un documento bien formado el siguiente:

```
1<editorial>
2  <libro>
3    <titulo>La vida está en otra parte
4    <autor>Milan Kundera</titulo>
5    </autor>
6  </libro>
7</editorial>
```

- *autor* debería cerrarse antes que *titulo* porque *autor* fue el último en abrirse

XML: Prólogo

- Si aparece, debe situarse al principio del documento
- Es obligatorio en XML 1.1 y opcional en XML 1.0
- Debe indicar la versión del idioma utilizado

```
<?xml version="1.0" ?>
```

- Opcionalmente, puede incluir los siguientes atributos:
 - *encoding*: tipo de codificación de caracteres utilizada
 - *standalone*: indica si el documento depende o no de otros documentos externos para comprobar la validez

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

XML: Documentos bien formados

- Para ayudarnos a crear documentos bien formados, las herramientas con utilidades específicas de XML nos señalan cuando hay un error.
- Por ejemplo, para Notepad++, podemos incorporarle el plugin de XML Tools
 - Menú Plugins → Administrar plugins → Buscamos XML Tools
 - Para saber más, podéis visualizar [el siguiente vídeo](#)