

Lenguaje de Marcas - XML

M04 - Lenguaje de Marcas
2021-2022

Alicia Vázquez
alicia.vazquez@itb.cat



XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ Relaciones y estructuras
 - ❑ Atributos
 - ❑ Declaración XML
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Espacio de nombres
-

XML (eXtensible Markup Language)

- ❑ XML (Lenguaje de Marcado eXtensible) es un lenguaje desarrollado por W3C (**World Wide Web Consortium**) que está basado en SGML.
- ❑ XML es un lenguaje utilizado para el **almacenamiento e intercambio de datos estructurados** entre distintas plataformas.
- ❑ XML es un **metalenguaje**, es decir, puede ser empleado para definir otros lenguajes, llamados dialectos XML. Por ejemplo, algunos lenguajes basados en XML son:
 - ❑ GML (Geography Markup Language).
 - ❑ MathML (Mathematical Markup Language).
 - ❑ RSS (Really Simple Syndication).
 - ❑ SVG (Scalable Vector Graphics).
 - ❑ XHTML (eXtensible HyperText Markup Language).

XML

- ❑ XML
 - ❑ **Elementos**
 - ❑ Comentarios
 - ❑ Relaciones y estructuras
 - ❑ Atributos
 - ❑ Declaración XML
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Espacio de nombres
-

ELEMENTOS

Los documentos XML están formados por **texto plano** (sin formato) y contienen **marcas** (etiquetas) definidas por el desarrollador. Dichas marcas, se encuentran entre los caracteres menor que "<", mayor que ">" y barra inclinada "/".

Ejemplo Si en un documento XML se quiere guardar el nombre *Elsa*, se puede escribir:

```
<nombre>Elsa</nombre>
```

ELEMENTOS: Sintaxis

<nombreEtiqueta> valor </nombreEtiqueta>

- **nombreEtiqueta:** Es el nombre que le damos a nuestra etiqueta/marca y deberá ser la misma al abrir que al cerrar.
- **valor:** Corresponde al valor que tiene esa etiqueta.

Ejemplo **<nombre>Elsa</nombre>**

→ Obsérvese que, entre la etiqueta de inicio (**<nombre>**) y la etiqueta de fin (**</nombre>**) se ha escrito el dato (valor) que se quiere almacenar. En este caso Elsa.

ELEMENTOS: Elementos vacíos

En un documento XML, un elemento puede no contener **ningún** valor. En tal caso hay que escribir:

<etiqueta></etiqueta>

Se puede expresar lo mismo escribiendo:

<etiqueta/>

Ejemplo Para escribir el elemento "nombre" vacío, se puede escribir:

<nombre></nombre> o <nombre/>

ELEMENTOS: Reglas para las etiquetas

El nombre de los elementos y atributos deben seguir unas reglas muy sencillas:

- ❑ Pueden contener letras del alfabeto (a..z, A..Z), números, puntos "", guiones medios "-" y guiones bajos "_". Las letras no inglesas (á, Á, ñ, Ñ...) están permitidas. Sin embargo, es recomendable no utilizarlas para reducir posibles incompatibilidades con programas que puedan no reconocerlas.
- ❑ El primer carácter tiene que ser una letra o un guión bajo "_" y **nunca** puede empezar por un número.
- ❑ Asimismo, pueden contener el carácter dos puntos ":". No obstante, su uso se reserva para cuando se definan espacios de nombres.
- ❑ Case Sensitive
- ❑ No pueden haber espacios o saltos de línea en medio de un nombre.

Ejercicio 1

Escribe correctamente el nombre de los elementos:

```
<xml:pais/>
<día>21</día>
<mes>10<mes/>
<ciudad>Barcelona</fciudad>
<Ciudad>Barcelona</ciudad>
<2colores>Naranja y verde</2colores>
<_rojo>
< Aficiones >Cine, Bailar, Nadar</ Aficiones >
<persona><nombree>Maria</persona></nombree>
<color favorito>blanco</color favorito>
```

ELEMENTOS: Contenido

El contenido de un elemento será todo lo que haya entre las etiquetas de apertura y de cierre. En el contenido podemos tener simplemente texto o bien puede ser otros elementos.

Ejemplo El elemento <persona> contiene dos elementos más: <nombre> y <apellido>, que además tienen contenido en su interior:

```
<persona>
    <nombre> Pedro </nombre>
    <apellido> Martí </apellido>
</persona>
```

ELEMENTOS: Contenido mixto

Una tercera posibilidad sería combinar los dos casos anteriores y que el contenido sea una mezcla de texto y elementos. **Elementos con contenido mixto**

```
<persona>
    Pere Martí
    <cargo> Director </cargo>
</persona>
```

El XML no define ninguna restricción a la hora de definir el contenido de los elementos.

- Podemos utilizar cualquier carácter representable con el código de caracteres.
- El contenido puede ser tan largo como nos haga falta.
- Se puede escribir en cualquier idioma del mundo.
- No importa que haya espacios en blanco o saltos de línea dentro del contenido.

XML

- XML
 - Elementos
 - Comentarios**
 - Relaciones y estructuras
 - Atributos
 - Declaración XML
 - Referencia a entidades
 - Documento XML bien formado
 - Espacio de nombres
-

COMENTARIOS

Los comentarios se incluyen los entre los símbolos <!-- y --> y se pueden poner comentarios en cualquier lugar del documento XML excepto dentro de las etiquetas.

<!-- Esto sería un comentario que no cuenta como elemento -->

@aliciaFPlnf

XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ **Relaciones y estructura**
 - ❑ Atributos
 - ❑ Declaración XML
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Espacio de nombres
-

RELACIONES

Un elemento (padre) puede contener a otro u otros elementos (hijos).

Ejemplo: El elemento "persona" contiene cuatro elementos (hijos): "nombre", "mujer", "fecha_de_nacimiento" y "ciudad". A su vez, el elemento "fecha_de_nacimiento" contiene otros tres elementos (hijos): "dia", "mes" y "anyo". Sólo el elemento "mujer" está vacío.

```
<persona>
  <nombre>Maria</nombre>
  <mujer/>
  <fecha_de_nacimiento>
    <dia>21</dia>
    <mes>10</mes>
    <anyo>1998</anyo>
  </fecha_de_nacimiento>
  <ciudad>Bilbao</ciudad>
</persona>
```

ESTRUCTURA

Todos los documentos XML válidos consiguen que la información que contiene se organice de manera jerárquica.

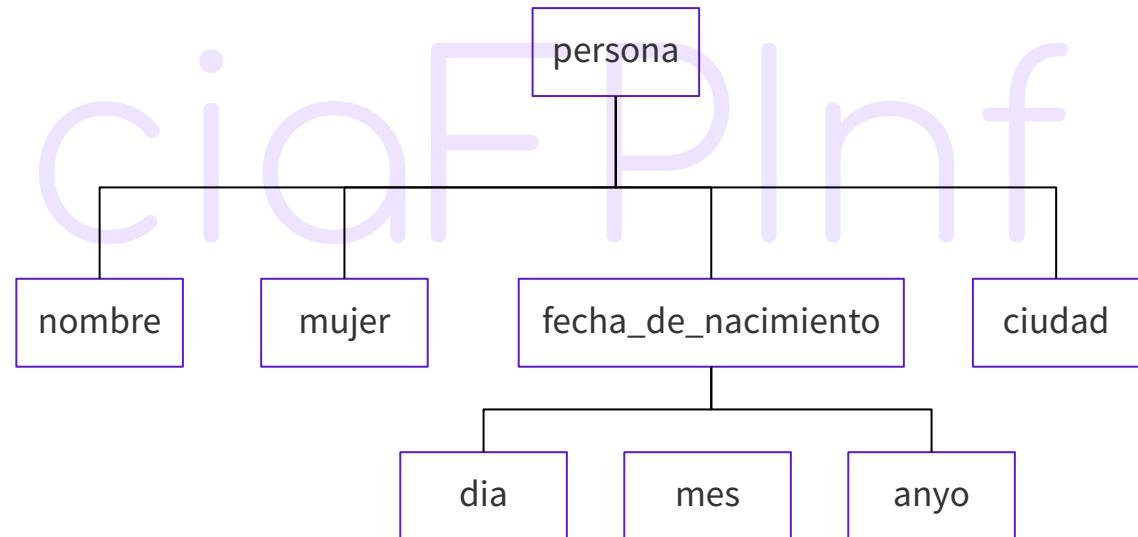
Los elementos que forman parte del contenido de un nodo se llaman **hijos**. Por equivalencia con las relaciones humanas a los hijos de los hijos de un elemento se denominan **nietos**.

Todos documento XML tiene que tener un **único elemento raíz (padre)** del que descienden todos los demás.

ESTRUCTURA: Árbol

De esta forma, **la estructura de cualquier documento XML se puede representar como un árbol invertido de elementos**. El árbol comienza en la raíz y las ramas para el nivel más bajo del árbol. Se dice que los elementos son los que dan estructura semántica al documento.

```
<persona>
  <nombre>Maria</nombre>
  <mujer/>
  <fecha_de_nacimiento>
    <dia>21</dia>
    <mes>10</mes>
    <anyo>1998</anyo>
  </fecha_de_nacimiento>
  <ciudad>Bilbao</ciudad>
</persona>
```



Ejercicio 2

Haz el árbol de la estructura que corresponda al siguiente XML

```
<Empresa>
  <Empleado>
    <Apellido>Perez</Apellido>
    <Nombre>Juan</Nombre>
    <NumEmpleado>1234567890</NumEmpleado>
    <Email>juanperez@gmail.com</Email>
    <Telefono>666 555 444</Telefono>
    <Direccion>
      <Calle>Carrer de Pau Claris, 121</Calle>
      <Ciudad>Barcelona</Ciudad>
      <CP>08009</CP>
    </Direccion>
  </Empleado>
</Empresa>
```

XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ Relaciones y estructura
 - ❑ **Atributos**
 - ❑ Declaración XML
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Espacio de nombres
-

ATRIBUTOS

Los elementos de un documento XML pueden tener atributos definidos en la etiqueta de inicio. **Un atributo sirve para proporcionar información extra sobre el elemento que lo contiene.**

<nombreElemento nombreAtributo = "valor">

- **nombreElemento:** Es el nombre del elemento al que pertenece el atributo.
- **nombreAtributo:** Es el nombre* del atributo que debe cumplir con las normas de nombre marcadas por XML.
- **valor:** Debe ir entre comillas dobles o simples.

*mismas reglas de definición que para las etiquetas.

ATRIBUTOS

El XML permite definir tantos atributos como sea necesario sin ningún tipo de restricción. Las únicas condiciones que debemos seguir es que cada uno de los atributos debe estar separado de los otros al menos por un espacio.

- El **orden** en que aparecen los atributos no tiene ninguna importancia.
- Los nombres de los atributos deben cumplir las mismas normas de **sintaxis** que los nombres de los elementos.
- No se pueden repetir los nombres de los atributos dentro de un mismo elemento, deben ser **únicos dentro de una elemento**.

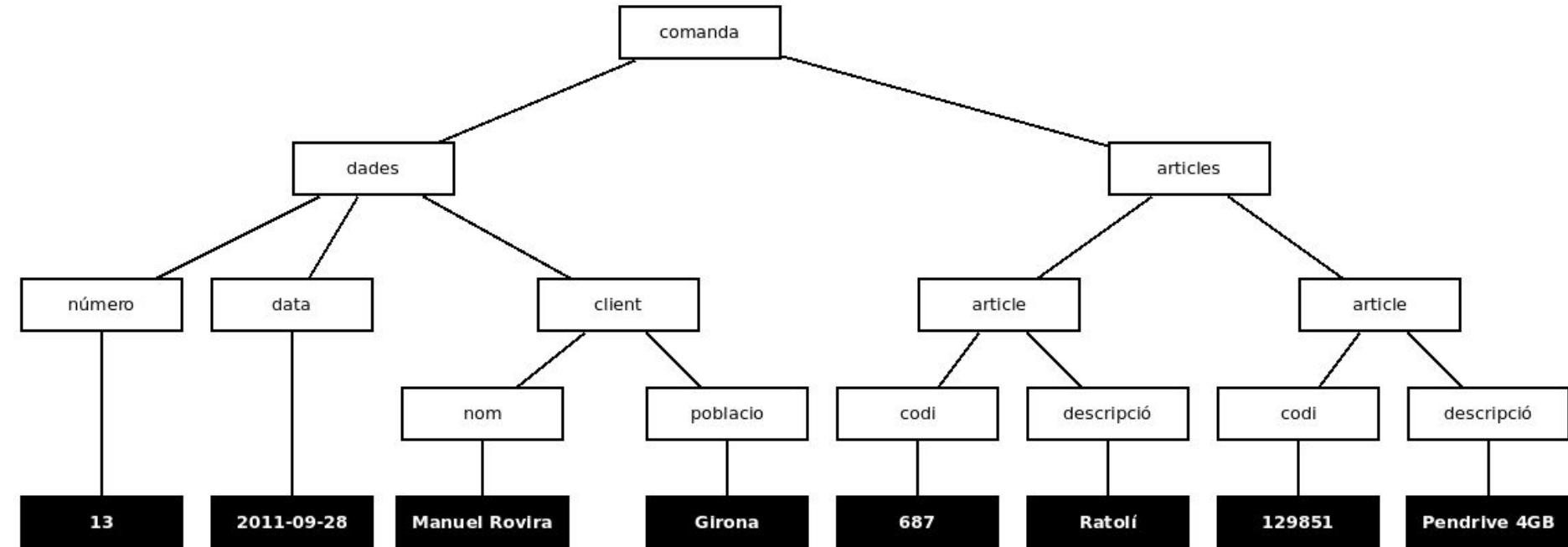
Ejercicio 3

Del ejercicio anterior modifica el código haciendo que **CP** sea tributo de **Ciudad** y **número de empleado** se atributo de **Empleado**.

¿Podría **Calle** ser atributo de **Dirección**? y **Ciudad**? Justifica tu respuesta.

Ejercicio 4

Escribe el XML que corresponde a este árbol **completo**.



ATRIBUTOS `xml:lang`

El atributo un **`xml:lang`** es un atributo **predefinido** en el lenguaje xml. Su cometido es **definir el idioma del contenido** de una etiquetas.

Ejemplo

```
<mensaje xml:lang="es">Hola</mensaje>
```

El atributo **`xml:lang`** se hereda a todos los hijos de la etiqueta

Ejemplo

```
<mensaje xml:lang="es">
    <saludo>Hola</saludo>
    <despedida>Adiós</despedida>
</mensaje>
```

Las etiquetas saludo y despedida también están en español.

Ejercicio 5

Corrige los siguientes fragmentos de código xml

```
<corredor posicio=1>Manel Roure</corredor>
<alumne delegat>Jaume Ravent</alumne>
<coche tipo combustible="gasolina">Seat Ibiza</coche>
<politico cargo="alcalde" cargo="diputat">Jordi Rufi</politico>
```

Escribe un código xml equivalente utilizando solo una etiqueta

```
<coche>
    <marca>Seat</marca>
    <color>Rojo</color>
    <matricula>B3456L</matricula>
    <conductor>Jorge</conductor>
    <conductor>Maria</conductor>
    <alquilado/>
</coche>
```

XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ Relaciones y estructura
 - ❑ Atributos
 - ❑ **Declaración xml**
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Espacio de nombres
-

DECLARACIÓN XML

El elemento **XML** es una etiqueta que contiene detalles que preparar un procesador XML para analizar el documento XML. Es opcional pero en caso de estar presente debe estar en el **comienzo** del documento.

```
<?xml version="versionNumber" encoding="encodingDeclaration"  
standalone="standaloneStatus"?>
```

Ejemplo

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

Normas declaración declaración XML

- Si la declaración XML está presente en el XML, se debe colocar en la primera línea del documento XML.
- Si la declaración XML está incluido, debe contener número de versión atributo.
- La declaración XML no tiene etiqueta de cierre i.e. </xml>

DECLARACIÓN XML

	Valores	Descripción
Versión*	1.0, 1.1	Versión del xml
Codificación	UTF-8, F-8, UTF-16, ISO-10646-UCS-2, ISO-10646-UCS-4, ISO-8859-1, etc	Código de caracteres en que se ha definido el texto.
Standalone	no, yes	Al escribir standalone="yes" se está indicando que el documento es independiente de otros, como por ejemplo de una DTD (Document Type Definition, Definición de Tipo de Documento) externa. En caso contrario, significará que el documento no es independiente.

Ejercicio 6

Corrige los siguientes fragmentos de código xml

```
<?xml encoding="UTF-8" standalone="yes"?>  
<perro></perro>
```

```
<?xml standalone="si"?>  
</?xml>
```

```
<gato></gato>  
<?xml version="1.0" encoding="UTF-8"?>  
<perro></perro>
```

XML

- XML
 - Elementos
 - Comentarios
 - Relaciones y estructura
 - Atributos
 - Declaración xml
 - Referencia a entidades**
 - Documento XML bien formado
 - Espacio de nombres
-

Caracteres reservados

En XML existen algunos caracteres que son especiales por su significado y, para escribirlos en un documento XML, se pueden utilizar las referencias a entidades mostradas en la siguiente tabla:

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

EJEMPLO

```
<?xml version="1.0" encoding="UTF-8"?>
<entidades>
    <menor_que>&lt;</menor_que>
    <mayor_que>&gt;</mayor_que>
    <comilla_doble>&quot;</comilla_doble>
    <comilla_simple>&apos;</comilla_simple>
    <ampersand>&amp;</ampersand>
</entidades>
```

```
▼<entidades>
    <menor_que><</menor_que>
    <mayor_que>></mayor_que>
    <comilla_doble>"</comilla_doble>
    <comilla_simple>'</comilla_simple>
    <ampersand>&</ampersand>
</entidades>
```

CDATA

Una sección **CDATA** es una etiqueta que comienza por **<![CDATA[** y termina por **]]>** y cuyo **contenido el procesador XML no interpreta como marcas sino como texto**. Es decir, que si aparecen los caracteres especiales (**< & " '**) en una sección **CDATA**, el procesador XML no interpreta que empieza una marca, sino que lo considera un carácter más.

Se suele utilizar en documentos en los que aparecen muchas veces esos caracteres especiales para no tener que estar utilizando las referencias a entidades (< y &) que dificultan la lectura del documento.

EJEMPLO

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
    <texto>Los caracteres < y & no pueden escribirse
    si no es como comienzo de marcas</texto>
</prueba>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
    <texto>Los caracteres &lt; y & no pueden
    escribirse si no es como comienzo de marcas
</texto></prueba>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<prueba>
    <texto><! [CDATA["Los caracteres < y & no pueden
    escribirse si no es como comienzo de marcas"]]>
</texto></prueba>
```

This page contains the following errors:

error on line 3 at column 26: StartTag: invalid element name

Below is a rendering of the page up to the first error.

Los caracteres

```
▼<prueba>
  ▼<texto>
    Los caracteres < y & no pueden escribirse si no es como
    comienzo de marcas
  </texto>
</prueba>
```

```
▼<prueba>
  ▼<texto>
    Los caracteres < y & no pueden escribirse si no es como
    comienzo de marcas
  </texto>
</prueba>
```

Ejercicio 7

Corrige los siguientes fragmentos de código xml y explica porqué no son válidos y cómo lo resuelves.

```
<persona>
    <nombre>Pere Garcia</nombre>
</persona>
```

```
<perro>
    <nombre>Rufy</nombre>
</perro>
```

```
<persona>
    <nombre>Pere</nombre>
    <apellido> Pérez </apellido>
</persona>
<persona>
    <apellido2>Garcia</apellido2>
    <edad>22</edad>
</persona>
```

XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ Relaciones y estructura
 - ❑ Atributos
 - ❑ Declaración xml
 - ❑ Referencia a entidades
 - ❑ **Documento XML bien formado**
-

Documento XML bien formado

Comprobar la **corrección** de un documento XML es comprobar que no se incluyen las reglas de definición del XML (vistas con anterioridad)

- Hay un solo elemento raíz que contiene al resto de elementos.
- Los nombres de las etiquetas tienen que seguir las reglas de validez.
- Las etiquetas de apertura, de cierre y vacías están correctamente anidadas (no se solapan) y no falta ni sobra ninguna etiqueta de apertura o cierre.
- Las etiquetas de cierre coinciden con las de apertura (incluso en el uso de mayúsculas y minúsculas).
- Las etiquetas de cierre no contienen atributos.
- Los valores de los atributos tienen que estar entre comillas (simples o dobles) y todos los atributos tienen algún valor.
- Ninguna etiqueta tiene dos atributos con el mismo nombre.



Documento XML bien formado

La manera más sencilla comprobar si un documento XML está bien formado es mediante el uso del navegador.

Existen otras herramientas por línea de comandos que permiten validar XML; libxml2 (GNU) , Apache Xerces (java), MSXML (microsoft)

The screenshot shows a web browser window with two tabs. The left tab displays the XML code:

```
<?xml version="1.0"?>
<holo>
  <adeu> asdasdasd </adeu>
</holo>
```

The right tab shows the validation results:

This page contains the following errors:

error on line 7 at column 17: Opening and ending tag mismatch: adeu line 0 and adeusssss

Below is a rendering of the page up to the first error.

Ejercicio 8

Haz un XML que guarde la siguiente información (un xml para cada ítem)

1. El alumno Juan García, nació el 23 de Febrero de 2001 en Barcelona.
2. Establecimientos: El Bar Pasqual es un bar de tapes de la avenida diagonal 232 propiedad de Joan Alcover. El restaurante Meravelles es un restaurant de menú en Meridiana 111 propiedad de Maria Garcia. El restaurant Racó de l'Avia és un restaurant de menú en Av. Roma 221 propiedad de Alba Puig.
3. Listado de obras de arte de un museo: Azul sobre blanco es un cuadro abstracto de Pere Àguila de 2014 hecho en óleo. Gris oscuro es un cuadro abstracto de Pere Àguila del 2010 hecho en acuarela. Camino largo es una escultura de Marta Lambert del 1981 de mármol.

Ejercicio 9: Recetas

Crea un xml que guarde información de recetas de cocina

- Como a mínimo tiene que haber dos recetas
- Como mínimo usa un atributo
- Como mínimo tiene que haber 4 niveles de etiquetas.
- Deben haber elementos repetidos.

XML

- XML
 - Elementos
 - Comentarios
 - Relaciones y estructura
 - Atributos
 - Declaración xml
 - Documento XML bien formado
 - Editores XML**
-

Editores XML

Un documento XML es un fichero de texto plano, por lo tanto cualquier editor de texto puede servir para crear y editar ficheros XML.

- ❑ **Editores de texto sencillos**

notepad, gedit, [vim](#), [jedit](#), [sublime text 3](#)

- ❑ **Editores con soporte XML**

[Eclipse](#) o el [Visual Studio Code](#) (plugins), [Android Studio](#)

- ❑ **Editores específicos XML**

[XML Copy Editor](#)

Escribir un libro (1)

Queremos escribir un libro usando el formalismo XML.

El libro está estructurado en:

- parte (al menos 2)
- capítulo (al menos 2)
- párrafo (al menos 2).

1. El libro debe contener la lista de autores (con apellido y nombre).
2. Todos los elementos deben tener un título, excepto el párrafo que contiene solo texto.
3. Proponed una estructuración XML de este documento (con 2 autores, 2 tema/parte, 2 capítulos por parte1 y 2 párrafos por capítulo).
4. Compruebe, con la ayuda del editor, que su documento está bien formado.

Advertencia: No utilizar atributos.

Índice general

I	Curso de matemáticas	= Parte	3
1.	Teoría		5
1.1.	Los números enteros		5
1.1.1.	Definición		5
1.1.2.	Ejemplos		5
1.2.	Fracciones		5
1.2.1.	Definición		5
1.2.2.	Ejemplos		5
2.	Ejercicios	= Capítulo	7
2.1.	Números enteros		7
2.2.	Fracciones		7
II	Curso de geometría		9
3.	Geometría del plano		11
3.1.	Definiciones		11
3.2.	Ejemplos		11
4.	Polígonos		13
4.1.	Definiciones		13
4.2.	Ejemplos		13

Escribir un libro (2)

1. Atributos

Queremos completar la estructura del documento XML del ejercicio anterior con los **atributos** de *nombre* y *apellido* de los autores y el *título* del libro, de las temáticas y de los capítulos.

Analizar la estructura del nuevo documento. ¿Hay posibles simplificaciones?

Comprueba, con la ayuda del editor, que su documento está bien formado.

2. Árbol

Haz el árbol de estructura de este último apartado con atributos.

XML

- ❑ XML
 - ❑ Elementos
 - ❑ Comentarios
 - ❑ Relaciones y estructura
 - ❑ Atributos
 - ❑ Declaración xml
 - ❑ Referencia a entidades
 - ❑ Documento XML bien formado
 - ❑ Editores XML
 - ❑ **Espacio de nombres**
-

ESPACIO DE NOMBRES

El espacio de nombres **namespace** es la forma que tiene el lenguaje xml para evitar **colisiones** de nombres entre elementos/etiquetas.

XML permite mezclar documentos; surge entonces el problema de etiquetas con el nombre repetido .

Por ejemplo: x3dom es un lenguaje basado en xml para definir entornos 3d.

<https://www.x3dom.org/>

Este lenguaje contiene la etiqueta <box>.

ESPACIO DE NOMBRES

Imaginemos que queremos definir nuestro propio lenguaje para guardar un listado de productos para e-commerce y queremos añadir una representación 3D del producto.

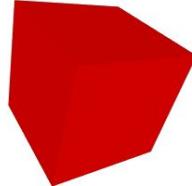
```
<producto>
  <nombre>Ordenador portátil</nombre>
  <precio>1000</precio>
  <box>de carton</box>
  <representacion3d>
    <x3d width='500px' height='400px'>
      <scene>
        <shape>
          <appearance>
            <material
diffuseColor='1 0 0'></material>
          </appearance>
          <box></box>
        </shape>
      </scene>
    </x3d>
  </representacion3d>
</producto>
```

La etiqueta box se repite en nuestro lenguaje y en el definido por 3dom, se crea una colisión!

<box>de carton</box>



<box></box>



Para nuestro lenguaje la etiqueta box se refiere a una caja de embalaje, para el lenguaje x3d significa una figura geométrica

ESPACIO DE NOMBRES

Para evitar este problema se añade al nombre de la etiqueta una cadena de caracteres única una URL. (*este código no es válido, es solo a nivel conceptual*)

```
<http://tienda.com:producto>
<http://tienda.com:nombre>Ordenador portàtil</http://tienda.com:nombre>
<http://tienda.com:precio>1000</http://tienda.com:precio>
<http://tienda.com:box>de carton</http://tienda.com:box>
<http://tienda.com:representacion3d>
  <https://www.x3dom.org:x3d width='500px' height='400px'>
    <https://www.x3dom.org:scene>
      <shape>
        <https://www.x3dom.org:appearance>
          <https://www.x3dom.org:material diffuseColor='1 0 0' />
        </https://www.x3dom.org:appearance>
        <></a><>https://www.x3dom.org:box>
      </shape>
    </https://www.x3dom.org:scene>
  </https://www.x3dom.org:x3d>
</http://tienda.com:representacion3d>
</http://tienda.com:producto>
```

ESPACIO DE NOMBRES

Para ganar en legibilidad se utiliza **alias** mediante el atributo **xmlns**.

```
<tienda:producto xmlns:tienda="http://tienda.com" xmlns:x3dom="https://www.x3dom.org">
  <tienda:nombre>Ordenador portátil</tienda:nombre>
  <tienda:precio>1000</tienda:precio>
  <tienda:box>de carton</tienda:box>
  <tienda:representacion3d>
    <x3dom:x3d width='500px' height='400px'>
      <x3dom:scene>
        <x3dom:shape>
          <x3dom:appearance>
            <x3dom:material diffuseColor='1 0 0'></x3dom:material>
          </x3dom:appearance>
          <x3dom:box></x3dom:box>
        </x3dom:shape>
      </x3dom:scene>
    </x3dom:x3d>
  </tienda:representacion3d>
</tienda:producto>
```

ESPAZIO DE NOMBRES

- ❑ **xmlns:prefijo="URI"**
- ❑ **xmlns="URI"** (espacio de nombres por defecto)
- ❑ **xmlns=""** (sin espacio de nombres)

Ejemplo

```
<e1:ejemplo xmlns:e1="http://www.miespacio.com/ejemplo1"  
           xmlns:e2="http://www.miespacio.com/ejemplo2">
```

Podemos especificar el espacio de nombres en cualquier elemento.

- **xmlns** es un atributo reservado para especificar el espacio de nombres.
- **prefijo:** corresponde al alias que le daremos para identificarlos y usarlo en el resto de etiquetas.
- **URI** (Uniform Resource Identifier, Identificador Uniforme de Recurso)

ESPACIO DE NOMBRES

Para ganar en legibilidad podemos definir un espacio por **defecto** y aprovechar la **herencia** entre elementos hijos.

```
<producto xmlns="http://tienda.com" xmlns:x3dom="https://www.x3dom.org">
  <nombre>Ordenador portàtil</nombre>
  <precio>1000</precio>
  <box>de carton</box>
  <representacion3d>
    <x3dom:x3d width='500px' height='400px'>
      <scene>
        <shape>
          <appearance>
            <material diffuseColor='1 0 0'></material>
          </appearance>
          <box></box>
        </shape>
      </scene>
    </x3dom:x3d>
  </representacion3d>
</producto>
```

ESPACIO DE NOMBRES más ejemplos

Imaginemos dos documentos XML con un listado de alumnos y profesores de la asignatura

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
<profesores>
    <nombre>Pilar Ruiz Pérez</nombre>
    <nombre>Tomás Rodríguez Hernández</nombre>
</profesores>
```

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<alumnos>
    <nombre>Fernando Fernández González</nombre>
    <nombre>Isabel González Fernández</nombre>
    <nombre>Ricardo Martínez López</nombre>
</alumnos>
```

Al hacer un documento sobre los miembros del curso DAW no se distinguiría los profesores de los alumnos, para resolverlo definiremos un espacio de nombres para cada contexto.

ESPAZIO DE NOMBRES más ejemplos

Para evitar colisiones de nombres definimos un espacio de nombre para cada contexto

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?>
<!DOCTYPE miembros>
<asistentes xmlns:alumnos="http://ASIR/alumnos"
  xmlns:profesores="http://ASIR/profesores">

  <alumnos:nombre>Fernando Fernández González</alumnos:nombre>
  <alumnos:nombre>Isabel González Fernández</alumnos:nombre>
  <alumnos:nombre>Ricardo Martínez López</alumnos:nombre>
  <profesores:nombre>Pilar Ruiz Pérez</profesores:nombre>
  <profesores:nombre>Tomás Rodríguez Hernández</profesores:nombre>
</asistentes>
```

Ejercicio 10

Evita la colisión de nombres en el siguiente fragmento de xml, conservando los nombres de las etiquetas y aplicando las normas vistas en los namespaces

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<inversion>
    <pais nombre="Francia">
        <capital>
            Paris
        </capital>
        <capital>
            1200M€
        </capital>
    </pais>
</inversion>
```

Referencias

Referencia Oficial: <https://www.w3schools.com/xml/default.asp>

Tutorial XML: <https://jorgesanchez.net/manuales/xml/introduccion-lenguajes-de-marcas.html>

Tutorial XML: <https://www.tutorialspoint.com/es/xml/>

Espacio de nombres: <https://www.abrirllave.com/xml/espacios-de-nombres.php>