Tema 8

Documents XML

Objectius

- Reconèixer la necessitat i els àmbits d' aplicació del llenguatge de marques XML.
- Conèixer les característiques pròpies del llenguatge de marques XML.
- Identificar les diverses tecnologies que formen la família de tecnologies XML.
- Comprendre l' estructura d' un document XML i les seves regles sintàctiques.
- Identificar els avantatges que aporten els espais de noms.
- Contrastar la necessitat de crear documents XML ben formats.

Introducció

Els temes anteriors s'han dedicat a l'estudi del llenguatge de marques HTML que s'utilitza per definir la presentació de pàgines i llocs Web. En aquest tema i en els temes següents s' avorta l' estudi del llenguatge de marques XML i les seves tecnologies associades. El llenguatge de marques XML serveix per definir què representa la informació que conté un document i sol emprar-se per crear documents destinats a emmagatzemar informació i a transmetre informació entre aplicacions informàtiques. Els següents temes aborden diversos aspectes més concrets relacionats amb l' ús del llenguatge de marques XML, així com amb la creació i utilització de documents XML mitjançant l' aplicació d' un conjunt de tecnologies associades.

Índex

8.1 LLENGUATGE DE MARQUES XML

El llenguatge de marques XML *(eXtensive Markup Language)* s'inclou en el tipus de llenguatges de marques orientats a la descripció. Aquest tipus de llenguatges permeten donar significat al document, tot i que no defineixen com s' ha de presentar visualment la informació continguda.

El llenguatge XML va ser concebut per descriure informació. Es tracta d' un llenguatge que permet definir un format d' emmagatzematge d' informació a base d' etiquetes o marques definides per l' usuari. A continuació, es pot apreciar un exemple d' aplicació del llenguatge XML per crear un document que conté una informació útil per a un propòsit concret:

```
<?xml version="1.0" encoding="utf-8" ?>
<mensatge>
  <remit>
    <nom>Juan Planelles Forner</nom>
    <email>juan.plaf@alicante.com</email>
  </remet>
  <destinatari>
    <nom>Bill Gates</nom>
    <email>president@microsoft.gov</email>
  </destinatari>
  <assumpte>Hola Bill</assumpte>
  <text>
      ¿Hola què tal? Fa molt que no m'escrius. A veure si truques
      i quedem per prendre alguna cosa.
  </text>
</missatge>
```

Habitualment, se sol dir que **el llenguatge de marques XML és un metallenguatge**. Això és degut al que permet als desenvolupadors definir els elements que es necessitin, així com l' estructura de la informació que millor convingui. De manera que, el llenguatge XML no disposa d' un conjunt de fix d' etiquetes i atributs, sinó que es tracta d' un format flexible per a l' emmagatzematge d' informació que pot ser adaptat a qualsevol camp d' aplicació. Per tant, és important tenir en compte que XML no és un llenguatge en si mateix, sinó que és un llenguatge estàndard que es pot utilitzar per crear llenguatges de marques que compleixin certs criteris. En altres paraules, XML descriu una sintaxi que s' utilitza per crear llenguatges de marques propis que estan adaptats a les necessitats específiques de cada aplicació concreta.

SGML, HTML i XML

SGML *(Standard Generalized Markup Language)* va ser creat com un llenguatge estàndard per al marcatge d'informació. Tanmateix, quan s' ha de tractar amb una gran quantitat d' informació complexa, el llenguatge SGML esdevé un llenguatge molt complicat que, a més, no resulta adequat per a l' intercanvi d' informació a través de la xarxa.

L'aplicació més coneguda de SGML és el llenguatge HTML *(HiperText Markup Language)*. L'objectiu del llenguatge HTML és definir la presentació visual de la informació continguda en un document sobre un navegador Web a través del servei d'informació Web.

El llenguatge de marques XML es va crear per facilitar l' intercanvi d' informació entre aplicacions informàtiques a través de la xarxa, utilitzant per a això un llenguatge de marques senzill. En aquest sentit, cal entendre que el llenguatge XML és un subconjunt de SGML, amb els mateixos objectius, però no tan complex. De fet, va ser dissenyat per ser compatible amb SGML, la qual cosa implica que qualsevol document que segueix les regles de sintaxi de XML és també un document SGML. Però tanmateix, lògicament, no tots els documents SGML són documents XML.

Per exemple, si es vol compartir informació sobre un nom, però també es vol utilitzar aquesta informació d'alguna altra manera concreta en un altre programa informàtic, en lloc de crear un arxiu de text amb el contingut següent:

```
<! DOCTYPE html>
<html>
<head>
    <title>Name</title>
    <meta charset="utf-8" />
</head>
<body>
    <P>John Doe</P>
</body>
</html>
```

Es pot crear un arxiu XML com el següent:

```
<?xml version="1.0" encoding="utf-8" ?>
<name>
  <first>John</first>
  <last>Doe</last>
</name>
```

Analitzant els dos exemples anteriors es pot observar que la informació continguda en ambdós casos està relacionada amb el nom d' una persona. Tanmateix, emprant el llenguatge XML la informació disposa d' una estructura: una dada anomenada first que correspon amb el nom i una altra anomenada last que correspon amb el cognom. En efecte, es pot observar que el document XML de l' exemple anterior incorpora elements que defineixen el significat de la informació emmagatzemada. Aquesta particularitat de dotar d' estructura i significat la informació emmagatzemada en el document són algunes de les característiques principals que aporta el llenguatge XML, la qual cosa, en determinades aplicacions, pot proporcionar certs avantatges en el desenvolupament de processaments d' emmagatzematge i transmissió de la informació a través d' Internet.

Avantatges de la utilització del llenguatge XML

Alguns dels principals avantatges de l' aplicació del llenguatge de marques XML són els següents:

- És directament utilitzable a Internet.
- Proporciona suport per a una àmplia varietat d' aplicacions per a transferència de dades.
- És compatible amb SGML.

- Permet crear documents XML llegibles i estructurats.
- Facilita crear un disseny ràpid i simple del llenguatge, encara que perfectament formalitzat.
- Permet crear documents XML.
- Permet representar informació de forma estructurada (Jeràrquica).
- Es basa en una gramàtica d' obligat compliment. Això facilita el desenvolupament de *parsers* i, per tant, promou la seva utilització massiva.
- Disposa de tecnologies que permeten definir les regles que ha de complir l' estructura interna que tindrà un document XML perquè sigui vàlid.

Exemple:

```
<?xml version="1.0" encoding="utf-8" ?>
<biblioteca>
  <llibre idioma ="anglès">
    <titulo>The Hobbit</titulo>
    <autor>J. R. R. Tolkien</autor>
    <editorial>Allen and Unwin</editorial>
  </llibre>
  <llibre idioma ="catellano">
    <titulo>El Quijote</titulo>
    <autor>Miguel de Cervantes</autor>
    <editorial>Alfaguara</editorial>
  </llibre>
  <llibre idioma ="catellano">
    <titulo>Harry Poter i la pedra filosofal</titulo>
    <autor>J.K. Rowling</autor>
    <editorial>Salamandra</editorial>
  </llibre>
</biblioteca>
```

En l' exemple anterior, es poden apreciar algunes de les característiques o avantatges de l' ús del llenguatge XML per a la creació de documents: estructura jeràrquica, senzillesa, llegibilitat, etc., així com el fet que s' hagi creat un llenguatge de marques específic per poder emmagatzemar la informació relativa als llibres d' una biblioteca.

8.2 FAMÍLIA DE TECNOLOGIES XML

El llenguatge XML posseeix un gran nombre de tecnologies associades que es complementen i ofereixen diverses funcionalitats específiques. L' ús del llenguatge XML s' aplica, principalment, per estructurar, emmagatzemar i intercanviar la informació. Aquests camps d' aplicació es corresponen amb temàtiques massa àmplies com per pensar que és possible abastar tota la seva complexitat mitjançant una única especificació. Per aquesta raó, s' han creat diverses especificacions

interrelacionades entre si que es complementen i que solen emprar-se en conjunt. Aquest conjunt d' especificacions constitueix la família de tecnologies XML.
A continuació, s' enumeren només les tecnologies XML més importants, algunes de les quals s' estudiaran en els propers temes:

- **XML 1.0 i 1.1**. Són les especificacions base a partir de les quals es construeix la família de tecnologies XML. Descriu la sintaxi i regles que han de complir els documents XML, així com les regles que han de complir els analitzadors XML. També defineix DTD, tot i que habitualment es tracta com una tecnologia XML diferent.

- **DTD *(Document Type Definition)***. És un llenguatge que permet especificar les regles, estructura i noms d' elements que han de complir els documents XML als quals s' vorin. És a dir, aquest llenguatge permet crear documents de validació per a arxius XML.

- **XML Schema**. La funció que compleix aquesta tecnologia XML és idèntica a l' anterior. Es tracta d'una altra tecnologia de validació XML. La diferència està en què els documents XML Schema posseeixen una sintaxi XML.

- **Namespacing**. Permet definir espais de noms. Un espai de noms és un mitjà per distingir entre un vocabulari XML i un altre, la qual cosa permet crear documents més consistents mitjançant l' ús de múltiples vocabularis dins d' un mateix document.

- **XPath**. És un llenguatge de consulta que permet seleccionar la informació continguda en un document XML.

- **XQuery**. Aquesta tecnologia compleix la mateixa funció que l' anterior. Permet consultar dades en els documents XML, manejant-los com si es tractar d' una base de dades.

- **CSS *(Cascade StyleSheet)***. Permet definir l' aparença de la presentació visual dels documents HTML, tot i que també es pot aplicar als documents XML.

- **XSL *(eXtensible Style sheet Languaje)***. Aquesta tecnologia permet definir el format de la presentació visual dels documents XML d' una manera adequada i específica. Ofereix moltes més possibilitats que CSS. Inclou la tecnologia XSLT que permet transformar o convertir un document XML per donar-li un determinat tipus de format de presentació.

- **DOM *(Document Object Model)***. Permet accedir a l' estructura jeràrquica d' un document XML, normalment per utilitzar-la des d' un llenguatge de programació.

- **SAX *(Simple API for XML)***. Permet l' ús d' eines per accedir a l' estructura jeràrquica d' un document XML a través d' un altre llenguatge. S'utilitza amb llenguatge Java.

- **XForms**. Permet definir formats de formularis per a la introducció de dades.

- **XLink**. Permet crear hipervincles en un document XML.

- **XPointer**. Semblant a l' anterior, permet especificar enllaços a elements externs.

8.3 CARACTERÍSTIQUES D' UN DOCUMENT XML

Els documents XML emmagatzemen informació estructurada que s' especifica mitjançant el llenguatge de marques XML. Les principals característiques d' un document XML són les següents:

- Es tracta d' un document estructurat, construït en un arxiu mitjançant text pla i l' extensió del qual sol ser .xml.

- L'aspecte general del document XML recorda al que té un document HTML, només que les etiquetes no són estàndard, sinó que han estat definides pel desenvolupador.

- Un document XML especifica contingut i estructura. El document està format, bàsicament, per una barreja d' informació de contingut i d' etiquetes de marcatge, tant d' obertura com de tancament. Les etiquetes de marcatge actuen sobre la informació de contingut del document i queden delimitades pels caràcters '<' i '>'.

- Un document XML no descriu cap aspecte relatiu a la presentació visual de la informació que conté.

El format d' un document XML és text pla o simple. Aquest format és adequat per estructurar, emmagatzemar i intercanviar la informació. Així que, mitjançant qualsevol editor de textos és possible crear un document XML. A més, els documents XML són eficients des del punt de vista de l' emmagatzematge, ja que la seva ocupació és, bàsicament, la que correspon a la informació continguda més la de les etiquetes que la delimiten.

Les etiquetes de marcatge d' un document XML són metainformació. Això és, informació relativa a la informació continguda en el document XML. Les etiquetes permeten definir l' estructura del document i en faciliten el processament, però, no són informació en si mateixa, ja que no aporten ni completen el contingut d' informació del document.

És important ressaltar que el llenguatge XML distingeix entre majúscules i minúscules, tant en les dades com en l' etiquetatge. També és convenient tenir en compte que, a diferència d' HTML, els espais en blanc usats en un document XML són significatius.

Bàsicament, un document XML es pot dividir en dues seccions:

- **Prologo**. Part que conté informació sobre el propi document XML, com pot ser la declaració de la versió de XML i altres declaracions, definicions i instruccions de processament.

- **Contingut**. Part del document XML que conté la informació pròpia del document a la qual se li ha afegit el marcatge. Obligatòriament, ha d' incloure un element arrel que envolta la resta d' elements.

Exemple:

```
<?xml version="1.0" encoding="utf-8" ?>
<novedades>
    <feta>Octubre 2016</data>
    <album genero="pop">
        <titulo>You Want It Darker</titulo>
        <interprete>Leonard Cohen</interprete>
    </album>
    <album genero="pop">
        <titulo>Indestructible</titulo>
        <interprete>Diego "El Cigala"</interprete>
    </album>
    <llibre>
        <titulo>Falcó</titulo>
        <autor>Arturo Pérez-Reverte</autor>
    </llibre>
    <llibre>
        <titulo>Perra negra</titulo>
        <autor>Timothy Snyder</autor>
    </llibre>
    <llibre>
        <titulo>El llibre dels miralls</titulo>
        <autor>Eugene Chirovici</autor>
    </llibre>
</novetats>
```

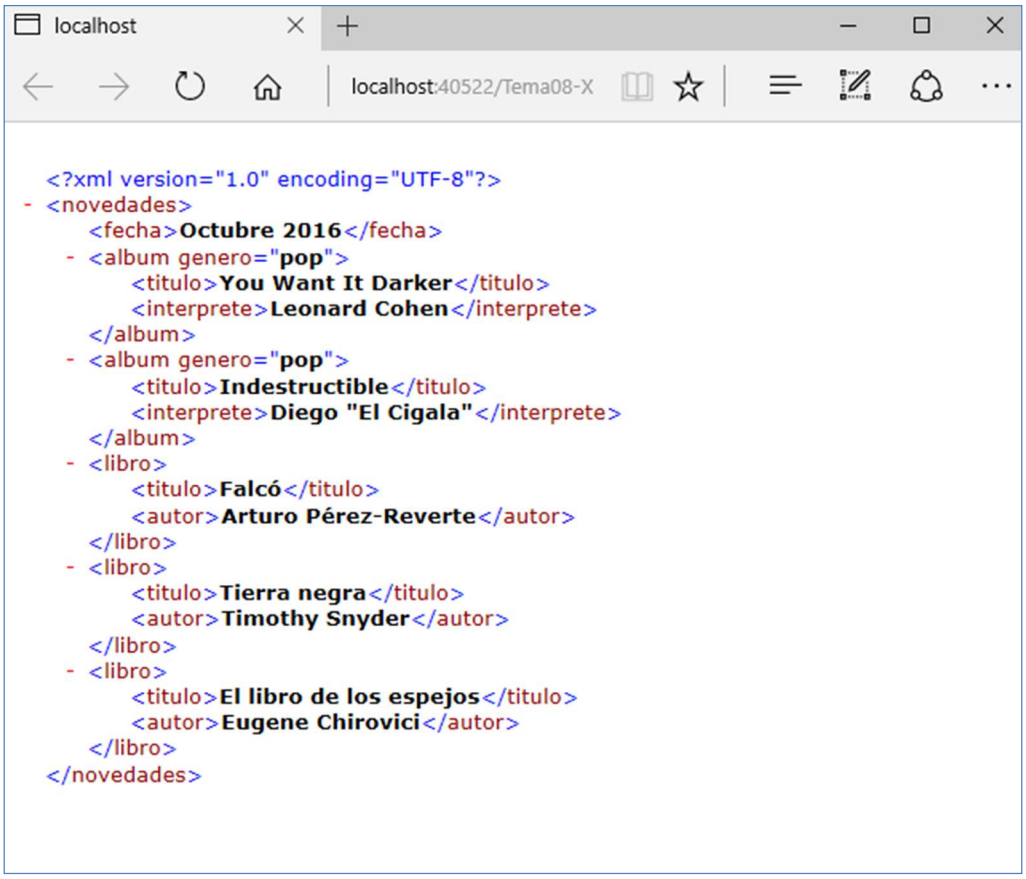
A continuació, s' analitza l' estructura del document XML corresponent a l' exemple anterior:

- En la primera línia del document, tenim el **pròleg** del document. En l'exemple anterior, el pròleg especifica la informació corresponent al tipus de document (xml) i la versió (1.0). A continuació, s' especifica el tipus de codificació de caràcters utilitzada en el document. Aquesta declaració és de caràcter opcional, tot i que és convenient incloure-la. El pròleg també pot incloure altres declaracions com, per exemple, la declaració de validació del document. Conforme es vagi avançant en l'estudi de XML, s'aniran incloent noves declaracions i instruccions de processament en el pròleg.

- La resta del document forma el **cos** del document XML que està format per diversos elements. En cada element, la informació de contingut del document queda delimitada per les etiquetes d' inici i de fi. Un document XML ha de tenir sempre un element arrel que conté tots els altres elements. En l'exemple, l'element arrel correspon amb l'etiqueta <novedades>.

Visualització d' un document XML

Els documents XML no disposen de cap visualització concreta en un navegador Web, ja que **un document XML no inclou cap aspecte relatiu a la presentació visual, sinó que tan sols inclou informació estructurada**. Quan un document XML ben format s'obre amb un navegador Web, llavors es mostrarà l'arbre de nodes del document, tal com es pot apreciar en la següent il·lustració.



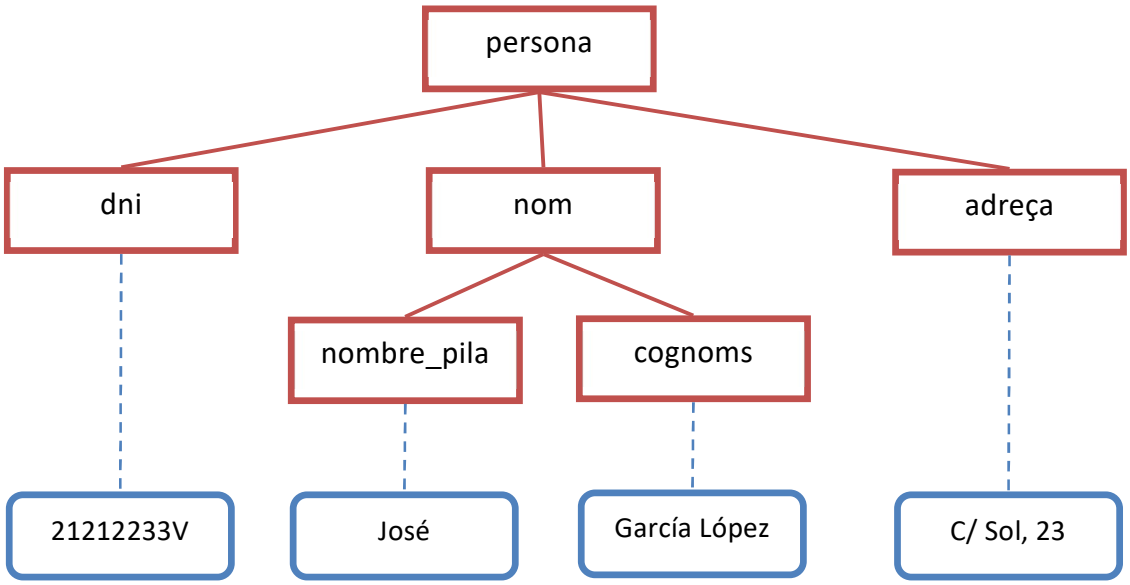
Existeixen diversos mètodes per representar visualment de forma més atractiva i agradable la informació continguda en un document XML. Per a això, es poden utilitzar fulls d' estil, els fulls de transformacions XSL i, també es pot utilitzar un llenguatge de programació lògica per construir programes que siguin capaços de processar adequadament un document XML. En els temes següents, s' estudiaran algunes d' aquestes tècniques.

8.4 ESTRUCTURA D' UN DOCUMENT XML

Un document XML té una estructura regular, però flexible (extensible) per poder estructurar la informació continguda de la manera més adequada.

Estructura jeràrquica de la informació

La informació continguda en un document XML s' organitza de forma estrictament jeràrquica, de manera que els elements o nodes del document es relacionen entre si mitjançant relacions de pares, fills, germans, ascendents, descendents, etc. A aquesta estructura jeràrquica se l' anomena arbre de nodes del document XML. A les parts de l' arbre que tenen fills se' ls denomina nodes intermedis o branques, mentre que a les parts que no tenen se' ls denomina nodes finals o fulles.



L' arbre de nodes anterior es correspon amb el següent document XML:

```
<?xml version="1.0" encoding="utf-8" ?>
<persona>
    <dni>21212233V</dni>
    <nom>
       <nombre_pila>José</nombre_pila>
       <apellidos>García López</cognoms>
    </nom>
    <direccion>C/ Sol, 23</direccion>
</persona>
```

En l' exemple es pot apreciar que l' element <persona> és l' element arrel. L'element <nom> és pare de l'element <apellidos> i, al seu torn, és fill de l'element <persona>. A més, els elements <dni>, <nom> i <direcció> són tots germans entre si i, alhora, són fills de l'element <persona>. També es pot observar que el contingut de les etiquetes també es considera com un node fill d' un element full. Per exemple, la informació "José" és fill de l'element <nombe_pila>.

Els documents XML han de seguir una estructura jeràrquica en relació amb l' ús de les etiquetes que delimiten els seus elements. Una etiqueta ha d' estar inclosa de manera correcta considerant l' organització jeràrquica de la informació. A més, les etiquetes de tancament dels elements amb contingut s' han d' aplicar de manera correcta segons l' ordre adequat de la seva etiqueta d' obertura.

Element arrel

Per sobre de qualsevol element, considerant l' estructura jeràrquica del document, se situa l' element o node arrel. Aquest node haurà de contenir, obligatòriament, la resta d' elements de document. Es tracta d' un element únic que actua com a punt de partida per recórrer l' arbre de nodes de XML d' un document i per ubicar la resta dels nodes.

Elements

Els elements constitueixen la unitat bàsica d' un document XML. Es tracta de contenidors d'informació que s'utilitzen per delimitar els diferents nodes que formen un document XML. Un element consta d'una etiqueta d'inici <NombreElement>, un contingut i una etiqueta de tancament </NombreElement>. El nom d' un element ha de ser un nom XMl vàlid. Els noms XML han de començar per una lletra o el caràcter subratllat, seguit per lletres, dígits, el caràcter punt, el caràcter guió o el caràcter de subratllat. A més, un nom XML no haurà de començar per la cadena XML, ja sigui en majúscules o en minúscules, ni pot contenir espais. Segons el contingut, els elements es classifiquen en:

- **Elements de contingut simple**. Són aquells elements que només contenen text. Per exemple, l'element <apellidos> en l'exemple anterior.

- **Elements que contenen altres elements**. Són aquells elements que posseeixen fills que són, al seu torn, altres elements. Per exemple, l'element <nom> en l'exemple anterior.

- **Elements de contingut combinat**. Són elements que contenen text i altres elements. En l' exemple anterior no hi ha cap cas d' aquest tipus. Per exemple:

  <parent>this is some text in my element</parent>

  En aquest exemple l'element <parent> té tres fills: un element fill que conté el text "this is some", un element fill i un altre element fill que conté el text "in my element".

- **Elements sense contingut**. Són elements que s'obren i tanquen amb una única etiqueta. El seu ús no és gaire habitual. Aquest tipus d' elements poden tenir també atributs. Per exemple: <separador />.

Atributs

Un atribut representa una informació complementària associada a un element. Es tracta de parells nom-valor que permeten especificar informació addicional d' un element. Apareixen a l' etiqueta d' inici o a la d' element buit. Cada element pot tenir una llista d' atributs associada, en la qual l' ordre és intranscendent però no poden aparèixer atributs repetits. El nom d' un atribut, igual que el d' un element, s' ha de tractar d' un nom XML vàlid. Per assignar un valor a un atribut s' utilitza el signe igual i el valor apareix entre cometes simples o dobles, seguint la següent sintaxi:

<element atribut1="valor1" atribut2="valor2">Contingut</element>

Per exemple:

```
<pès unitat="grams" 0,01">5,73</pes>
```

Identificació dels elements

Quan es crea un document XML, es recomana començar identificant quins seran els elements que hi apareixeran. Una determinada descomposició en elements pot ser més o menys adequada depenent de l' ús que es vagi a fer del document.

Supose que es pretén incloure el nom d' una persona en un document. Es plantegen diverses alternatives. Per exemple:

```
<nom>Juan Martín Fernández Moreno de la Vega </nom>
```

O bé:

```
<nom>
  <nombre_pila>Juan Martín</nombre_pila>
  <apellidos>Fernández Moreno de la Vega</apellidos>
</nom>
```

O bé:

```
<nom>
  <nombre_pila>Juan Martín</nombre_pila>
  <primer_apellido>Fernández</primer_apellido>
  <segundo_apellido>Moreno de la Vega</segundo_apellido>
</nom>
```

En aquests casos, Quina seria la millor solució? La resposta més correcta a aquesta pregunta és que depèn de la funcionalitat que hagi de tenir el document. Cadascuna de les estructures XML de l' exemple anterior proporciona unes o altres prestacions, essent totes elles correctes. En el cas que sempre s'hagi de treballar amb el nom com un tot, seria recomanable la primera opció. Però si es necessita processar els cognoms i el nom de pila per separat, serien recomanables qualsevol de les altres dues opcions. Per prendre la decisió adequada cal tenir una idea preconcebuda del processament que es realitzarà sobre el document XML en aquest moment i en un futur. En general, se sol considerar que es podrà accedir a un ús que proporcioni major funcionalitat, com més estructurada estigui la informació continguda en un document XML.

Ús d' elements o d' atributs

La informació continguda en un document XML pot aparèixer, tant en el contingut dels elements, com en el valor dels atributs. En ambdós casos es representa un text que constitueix la informació de contingut del document. Una pregunta que es pot plantejar és: Quan s'han d'utilitzar elements i quan atributs? Per respondre a aquesta pregunta s' analitzen els següents exemples, que es refereixen a documents XML que emmagatzemen informació relativa a les diapositives que formen part d' una determinada presentació. El primer exemple és:

```
<?xml version="1.0" encoding="utf-8" ?>
<presentació>
   <diapositiva>
      <ordre>1</ordre>
      <títol>Introducció</títol>
      <tiempo_exposicion>mei</tiempo_exposicion>
   </diapositiva>
   <diapositiva>
```

```
      <ordre>2</ordre>
      <títol>Ús d'elements o tributs</títol>
      <tiempo_exposicion>poc</tiempo_exposicion>
   </diapositiva>
  ...
</presentació>
```

En el segon exemple s' ha utilitzat un atribut per especificar el títol de la diapositiva:

```
<?xml version="1.0" encoding="utf-8" ?>
<presentació>
   <diapositiva títol="Introducció">
      <ordre>1</ordre>
      <tiempo_exposicion>mei</tiempo_exposicion>
   </diapositiva>
 <diapositiva títol ="Ús d'elements o tributs">
      <ordre>2</ordre>
      <tiempo_exposicion>poc</tiempo_exposicion>
   </diapositiva>
  ...
</presentació>
```

Ambdós exemples són equivalents des del punt de vista de la informació continguda, però El títol de la transparència s'hauria de crear com a element o com a atribut? I el temps d'exposició de cada diapositiva? En general, els següents criteris poden ajudar a resoldre aquestes qüestions o altres similars que poden plantejar-se:

- Si la informació té una estructura interna ha de ser un element.
- Si conté una gran quantitat d' informació, sembla més adequat utilitzar un element.
- En alguns casos, sol aplicar-se un criteri lingüístic: un substantiu sol convertir-se en element i un adjectiu en un atribut.
- Aquella informació que hagi de tenir un processament complex ha de ser un element.

Aplicant les pautes anteriors es pot concloure que el títol de la diapositiva hauria de ser un element, ja que té caràcter substantiu i, possiblement, podria requerir ser processat d' alguna manera. Tanmateix, el temps d'exposició de cada diapositiva, que pot prendre els valors poc, mitjà i molt; no té caràcter substantiu i no té perquè ser processada, per tant, podria ser perfectament un atribut. Com a consell final, en cas de dubte, s' ha d' utilitzar un element.

Caràcters especials

S' utilitzen entitats predefinides per representar caràcters especials de marcatge. Quan s' utilitza el llenguatge XML existeixen diversos caràcters que tenen un significat especial en els documents XML. Aquests caràcters són: &, <, >, ' (comilla simple) i " (cometes dobles).

Si es requereix incloure un d' aquests caràcters dins del contingut d' informació del document XML s' haurà d' utilitzar l' entitat predefinida corresponent segons la taula següent.

Entitat	Caràcter

&	&
<	<
>	>
'	'
"	"

El següent exemple especifica que el contingut de l'element <libreria> sigui "Barnes & Noble".

```
<libreria>Barnes & Noble</llibreria>
```

Seccions CDATA

Les seccions CDATA, de l' anglès *character data* que significa dades formades per caràcters, són conjunts de caràcters que no s' analitzen per part del processador XML. En aquest sentit, actua com si es tractés d'un comentari. L' especificació d' aquestes seccions permet agilitar l' anàlisi del document i permet al desenvolupador incloure-hi els caràcters especials & i <. Les seccions CDATA no poden aparèixer abans de l' etiqueta d' obertura de l' element arrel ni després del seu tancament. Una secció CDATA s'inicia amb la cadena **<![CDATA[** i acaba amb **]]>** i fa que l'analitzador sintàctic de XML interpreti el text que conté com una cadena de caràcters i no com a contingut etiquetat. Per exemple:

```
<lista>
  <nom>John Doe</nom>
  <correo> <! [CDATA[<jdoe@server.com>]]> </correu>
</llista>
```

En general, es poden presentar els casos següents:

- Els caràcters especials < i & sempre s' interpreten com a especials, excepte en seccions CDATA i en les seccions de comentaris.
- Només es pot incloure la cadena]]> com a indicació de final d'una secció CDATA. Si es necessita usar aquesta cadena com a contingut, caldrà utilitzar les entitats predefinides.
- En general, els caràcters ' (comilla simple) i " (comilla doble) no poden aparèixer en els valors que utilitzin com a delimitador el mateix tipus de cometes. Se sol utilitzar com a delimitador de valor el caràcter " i el caràcter ' en el contingut, si fos necessari.

Comentaris

Les seccions de comentari en XML segueixen la mateixa sintaxi que en HTML, comencen amb la seqüència **<!--** i acaben amb la seqüència **-->**. Per exemple: <!-- Això és un comentari -->.

Instruccions de processament

Les instruccions de processament són instruccions que van dirigides al programa que processa el document XML i, per tant, depenen del processador de XML que s' utilitzi en cada cas. Aquestes instruccions no formen part del contingut del document, de fet, s' especifiquen en la part de pròleg del document. S' utilitzen per proporcionar informació al processador XML.

Les instruccions de processament comencen per la seqüència de caràcters <?. I finalitzen amb la seqüència ¿>. Per exemple, la declaració del tipus de codificació de caràcters i la versió de XML usada en el document es defineix mitjançant una instrucció de processament, de la manera següent:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Espais de noms o *namespaces*

És un mecanisme de processament que s'empra per **evitar conflictes de noms**. D' aquesta manera, és possible distingir elements o atributs en un mateix document XML, encara que tinguin idèntics noms, però diferents definicions. La coincidència de noms pot ocórrer quan s' empren diversos vocabularis per estructurar la informació en un mateix document XML. Els espais de noms es declaren mitjançant l' atribut xmlns dins l' etiqueta d' obertura de l' element en qüestió, d' acord amb la sintaxi següent:

<prefix:nombre_elemento xmlns:prefix="URI_del_espacio_de_nombres">

El prefix de l' espai de noms es pot especificar utilitzant qualsevol cadena de text. Un cop declarat el prefix, ja es pot utilitzar per qualificar els elements i atributs d' un document XML i per associar-los a l' identificador URI de l' espai de noms. Com que el prefix d' un espai de noms afecta tot el document, es recomana utilitzar un nom curt. L' identificador URI de l' espai de noms ha de ser un valor de cadena únic, encara que, en realitat, no es comprova de cap manera mitjançant cap connexió. Cal tenir en compte que l' adreça URI no és més que el nom lògic de l' espai de noms que actua com un identificador únic. Per exemple:

```
<?xml version =" 1.0" encoding =" UTF-8"? >
<e1:exemple xmlns:e1="http://www.abrirllave.com/ejemplo1"
            xmlns:e2="http://www.abrirllave.com/ejemplo2">
   <e1:carta>
      <e1:pal>Corazones</e1:palo>
      <e1:numero>7</e1:numero>
   </e1:carta>
   <e2:carta>
      <e2:carnes>
         <e2:filete_de_ternera preu="12.95"/>
         <e2:solomillo_a_la_pimienta preu="13.60"/>
      </e2:carns>
      <e2:00>
         <e2:lenguado_al_horno preu="16.20"/>
         <e2:merluza_en_salsa_verde preu="15.85"/>
      </e2:pescats>
  </e2:carta>
</e1:exemple>
```

En l' exemple anterior es pot observar observa que l' atribut xmlns s' utilitza en l' etiqueta d' obertura de l' element <empl> i, que s' han definit dos espais de noms que fan referència a les URI:

http://www.abrirllave.com/ejemplo1 i http://www.abrirllave.com/ejemplo2. També es pot observar que els prefixos e1 i e2, respectivament, s' han afegit a les etiquetes corresponents que apareixen en el document XML.

En un document XML, els espais de noms es poden definir en l' element arrel, com es pot comprovar en l' exemple anterior. O també, directament en els elements que els hagin d' utilitzar, com es pot comprovar en l' exemple següent:

```
<?xml version="1.0" encoding="utf-8" ?>
<e1:exemple xmlns:e1="http://www.abrirllave.com/ejemplo1">
    <e1:carta>
      <e1:pal>Corazones</e1:palo>
      <e1:numero>7</e1:numero>
    </e1:carta>
    <e2:carta xmlns:e2="http://www.abrirllave.com/ejemplo2">
       <e2:carnes>
           <e2:filete_de_ternera preu="12.95"/>
           <e2:solomillo_a_la_pimienta preu="13.60"/>
       </e2:carns>
       <e2:00>
           <e2:lenguado_al_horno preu="16.20"/>
           <e2:merluza_en_salsa_verde preu="15.85"/>
       </e2:pescats>
    </e2:carta>
</e1:exemple>
```

Es poden definir espais de noms per defecte. D'aquesta manera, tant l'element on s'ha definit l'espai de noms, com tots els seus successors (fills, fills de fills, etc.), pertanyeran a aquest espai de noms. En efecte, un espai de noms és efectiu des del moment de la seva declaració fins a la fi de l' element en què s' ha declarat. Per exemple:

```
<?xml version="1.0" encoding="utf-8" ?>
<empl xmlns="http://www.abrirllave.com/ejemplo1">
   <carta>
      <palo>Corazones</palo>
      <numero>7</numero>
   </carta>
   <carta xmlns="http://www.abrirllave.com/ejemplo2">
      <carnes>
         <filete_de_ternera preu="12.95"/>
         <solomillo_a_la_pimienta preu="13.60"/>
      </carns>
      <pescats>
         <lenguado_al_horno preu="16.20"/>
         <merluza_en_salsa_verde preu="15.85"/>
      </pescats>
   </carta>
</exemple>
```

En l' exemple anterior, inicialment es defineix un espai de noms per defecte per a l' element <empl> i els continguts en ell. Després, es defineix un altre espai de noms, que per defecte afecta l' element <carta> i els seus successors: <carnes>, <pescats>, <filete_de_ternera>, etc.

8.5 DOCUMENT BEN FORMAT. *PARSERS*

L' especificació XML defineix la sintaxi del llenguatge quant a: la forma d' utilitzar les etiquetes per delimitar els elements, el format de les etiquetes, les normes dels noms dels elements i atributs i, finalment, la posició dels atributs.

Un document XML, es diu que està **ben format** si compleix les regles establertes pel W3C en les especificacions per a XML. Algunes de les principals regles de les especificacions de XML són les següents:

- El document XML pot començar per una instrucció de processament xml en el pròleg, que indica la versió de XML i, opcionalment, la codificació de caràcters mitjançant l' atribut *encoding*. La codificació de caràcters per defecte és UTF-8. En aquesta mateixa declaració es pot incloure l' atribut *estàndard*, que especifica si el document es processa independentment o requereix d' altres arxius externs per a això. El valor per defecte de l' atribut *estàndard* és no. Exemples:

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

```
<?xml version =" 1.0" encoding =" iso-8859-1"? >
```

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

- Un document pot incloure altres declaracions, definicions i instruccions de processament en el pròleg. Per exemple, opcionalment, pot incloure una declaració de tipus DTD que serveix per validar el document XML.

- Ha d' existir un únic element arrel. L' element arrel tindrà com a descendents tots els altres elements.

- Els elements que no siguin buits han de tenir una etiqueta d' obertura i una altra de tancament. I aquells que siguin buits s'han de tancar amb />, Exemple: <prova />.

- Els elements han d' aparèixer correctament anitats quant a la seva obertura i el seu tancament, és a dir, no es poden solapar. Els elements s' han de tancar en ordre invers a com s' obren, seguint les característiques específiques d' una estructura jeràrquica estricta.

- Els noms dels elements i atributs són sensibles a majúscules i minúscules.

- Els valors dels atributs han d' aparèixer entre cometes simples o dobles.

- No hi pot haver dos atributs amb el mateix nom associats al mateix element.

- No es poden introduir ni instruccions de processament ni comentaris enlloc del contingut dels elements, que està delimitat per les etiquetes d' obertura i tancament.

- No poden aparèixer els caràcters < ni & en el contingut textual dels elements ni dels atributs.

- No hi pot haver res escrit abans de la instrucció de processament: <?xml ...? >.

Una forma de comprovar que un document XML està ben format és obrint-lo amb un navegador Web. Si el document XML està ben format, llavors mostrarà l'arbre de nodes.

La importància que revesteix el fet que un document XML estigui ben format deriva del fet que un document ben format pot ser analitzable sintàcticament. I, per tant, en complir les normes de l' especificació XML podrà ser processat mitjançant els programes adequats.

Parsers o analitzadors sintàctics

Un programa *parser* és un programa capaç d'analitzar sintàcticament un document XML. Per aquest motiu també solen denominar-se analitzadors sintàctics.

Existeixen multitud de *parsers* o analitzadors sintàctics que han estat construïts emprant diversos llenguatges de programació. Els *parsers* de XML són capaços de detectar els errors sintàctics o estructurals d' un document XML, és a dir, detecten si està o no ben format. I, a continuació, poden notificàs al programa que processa el document XML per realitzar les tasques que siguin requerides. Aquesta funcionalitat és important per a un desenvolupador, ja que l' allibera de la tasca de detectar els errors sintàctics de XML.

Alguns *parsers*, a més, van més enllà de la simple capacitat de detectar si un document està ben format i són capaços d'identificar si el document XML és vàlid.

Document ben format i document vàlid

No s' ha de confondre que un document estigui ben format, amb què un document sigui vàlid, es tracta de dos conceptes diferents:

- **Document ben format**. Tal com s' ha vist anteriorment, un document XML està ben format si és conforme a una sèrie de regles bàsiques i generals, que estan descrites en l' especificació XML i que són aplicables a qualsevol document XML.

- **Document vàlid**. Un document XML és vàlid si compleix amb una sèrie de regles especificades per l'usuari, com, per exemple: quines etiquetes poden aparèixer en el document, en quin ordre han d'aparèixer, com es poden anellar, etc. Aquesta és la manera com es pot utilitzar el llenguatge XML per definir la sintaxi d' un llenguatge propi. S' entén per document XML vàlid aquell en el qual l' estructura, la posició i el nombre d' etiquetes compleixen amb les regles de validació especificades i, a més, el contingut d' informació té sentit.

Un cop s' ha comprovat que un document està ben format, es pot comprovar si és vàlid. Perquè un document XML pugui ser processat haurà d' estar ben format i ser vàlid.

A continuació, s' analitza el següent codi que forma part d' un document XML.

```
<Ingredient>
    <Cantitat unitat="peça">3</Quantitat>
```

```
    <Cantitat unitat="litre">4</Quantitat>
    <Item>Patatas</Item>
</Ingredient>
```

El codi XML de l' exemple anterior està ben format, perquè compleix totes les normes de l' especificació XML. Tanmateix, pot mancar de sentit semàntic, és a dir, la informació que conté pot no tenir un significat adequat. Es pot observar que s' ha definit dues vegades la quantitat de patates a utilitzar en la recepta. A més, una de les vegades s'ha definit una quantitat de patates de 4 litres. El problema de l'exemple anterior és que el document XML està ben format, però no té utilitat, ja que la informació que proporciona no té sentit. Per aquest motiu, cal poder comprovar que un document XML tingui significat. En aquest cas, s' hauria de comprovar que cada ingredient tingui definida només una etiqueta de tipus quantitat i que aquesta tingui un atribut opcional adequat a l' ingredient. Per a això, la família de tecnologies XML incorpora diversos llenguatges específics que permeten comprovar l' estructura del document XML per poder determinar si un document XML és vàlid. En el proper tema s'estudia un d'aquests llenguatges de validació que és: DTD *(Document Type Definition)*.

8.6 PROCEDIMENT PER A LA CREACIÓ D' UN DOCUMENT XML

El procés de creació d' un document XML pot ser el següent:

1. Conèixer quina informació s' inclourà en el document.
2. Obtenir el diagrama jeràrquic d' estructura del document:
    - Organitzar la informació mitjançant elements com un arbre de relacions jeràrquiques pare/fill. Sempre cal tenir en compte que el llenguatge de marques XML té una estructura jeràrquica.
    - Identificar aquella informació que es repeteix.
    - Identificar els elements que només incorporen informació textual en el seu contingut i aquells elements que, al seu torn, incorporen a altres elements.
    - Diferenciar elements i atributs. En general, podrà ser un atribut aquell element d' informació que no sigui rellevant o que no contingui a altres elements.
3. Crear, codificar i provar el document XML.
4. Comprovar que el document està ben format.
5. Comprovar que el document és vàlid.

Glossari de termes

Metainformació, o metadades. És "informació sobre la informació". Són descripcions estructurades i opcionals que estan disponibles de forma pública per ajudar a identificar, descobrir, valorar i administrar la informació continguda en un objecte. És a dir, es tracta d'informació no rellevant per a l'usuari final però sí de summa importància per al sistema que maneja les dades. Les metadades són enviades al costat de la informació quan es realitza alguna petició o actualització d' aquesta.

Metallenguatge. En lingüística, és un llenguatge que es fa servir per parlar sobre un altre llenguatge o per descriure'l. Al llenguatge sobre el qual s' està parlant se l' anomena llenguatge objecte. El metallenguatge pot ser idèntic al llenguatge objecte, per exemple, quan es parla sobre l'espanyol fent servir l'espanyol mateix. Un metallenguatge pot ser, alhora, el llenguatge objecte d' un altre metalenguatge d' ordre superior, i així successivament. I també, diferents metallenguatges poden parlar sobre diferents aspectes d'un mateix llenguatge objecte. En un sentit més general, pot referir-se a qualsevol terminologia o llenguatge usat per parlar amb referència al mateix llenguatge. Per exemple, un text sobre gramàtica o una discussió sobre l'ús del llenguatge.

Parser XML. És un analitzador sintàctic de XML. És un programa informàtic que llegeix un document XML i verifica que estigui ben format, alguns també comproven que el codi XML sigui vàlid. El *parser*, analitzador o processador de XML és l' eina principal de qualsevol aplicació XML. Mitjançant l' ús d' un parser no solament es pot comprovar si els documents XML estan ben formats o són vàlids, sinó que també poden ser incorporar-los a altres aplicacions informàtiques, sempre que aquestes puguin manipular i treballar amb documents XML. Els parsers XML es poden classificar en dos grups principals: sense validació, el parser no valida el document utilitzant un mecanisme adequat per a això, sinó que només el document estigui ben format d'acord amb les regles de sintaxi de XML; i, amb validació: a més de comprovar que el document està ben format segons les regles, comprova que el document sigui vàlid.