Tema 2. XML: Emmagatzemament de dades

Índex de continguts

2.1. Què és XML?

XML o eXtended Markup Language és un sistema de emmagatzemament d'informació basat en marques o etiquetes definides per l'usuari. XML està dissenyat per a emmagatzemar dades i enviar-les per la xarxa. A diferència de HTML (un altre llenguatge de marques), XML no diu res de com mostrar les dades en un navegador. A XML les etiquetes no estan predefinides com a HTML. Cada usuari pot definir les seves etiquetes depenent de l'àmbit al que pertany el document.

Els llenguatges de marques han de complir una sèrie de regles. Aquestes regles faran que el document XML sigui correcte sintàcticament o estigui ben format. Per exemples, les etiquetes s'han d'obrir i tancar en ordre invers, els valors dels atributs han d'anar entre cometes, comentaris, etc.

Una vegada tenim el document ben format, podem validar que la informació que surt en el document XML és correcta. Per exemple, indicant quins elements i atributs poden aparèixer, quins elements son optatius i quins no. Per exemple,

podem definir un document XML per guardar els llibres d'una biblioteca, i definir un identificador únic (un número per exemple) per a cada llibre. En cas parlariem d'un document vàlid.

Hi ha dues tècniques de validació de documents XML: DTD i els esquemes XML (XSD). DTD és heratada de SGML i va ser molt popular al anys 2000. XSD és un mecanisme molt més potent i modern i és el que veurem en aquest mòdul.

2.2. Què és i què no és XML.

XML és un estandar o norma, no una implementació concreata d'un llenguatge. És un metallenguatge de marques, és a dir no defineix cap conjunt d'etiquetes fixe, sino que els desenvolupadors poden crear els elements que necessitin amb l'estructura específica.

És un format flexible que permet adaptar-se a l'aplicació que desitgem. Per exemple:

- Si volem fer un document per a la banca tindrem etiquetes com <interes>, <plaç>, <compte> o <desfalc>.
- Si volem fer un document per a League of Legends, tindrem etiquetes com <campio>, <habilitat>, <objecte> o <encis>.

Què NO és XML?

- **No és un llenguatge de programació**, de forma que no existeixen compiladors de XML que generin executables a partir d'un document XML.
- **No és un protocol de comunicacion**, de forma que no enviarà dades per nosaltres a Internet, com tampoc ho fa HTML. Tenim protocols de comunicacions com HTTP, FTP.
- **No és un gestor de BBDD (SGDB)**. Una BBDD relacional pot contenir camps de tipus XML. Existeixen inclús BBDD natives en XML, és a dir que guarden i recuperen la informació en format XML, però aquests sistemes en sí mateixos no són una BBDD.

XML té format de texte pla, i es pot transmetre per Internet doncs són relativament lleugers. L'extensió dels arxius serà XML (encara que no és obligatori, qualsevol arxiu XML es pot veure amb un editor de texte).

2.3. Estructura d'un document XML

La informació en un document XML s'organitza de forma jeràrquica, de forma que els elements es relacionen entre ells mitjançant relacions: pares, fills, germans.

Aquesta estructura jeràrquica es denomina arbre. Cada element o node està connectat amb altre node. Als nodes que no tenen fills se'ls anomena nodes finals o fulles i a la resta nodes intermitgos o branques.

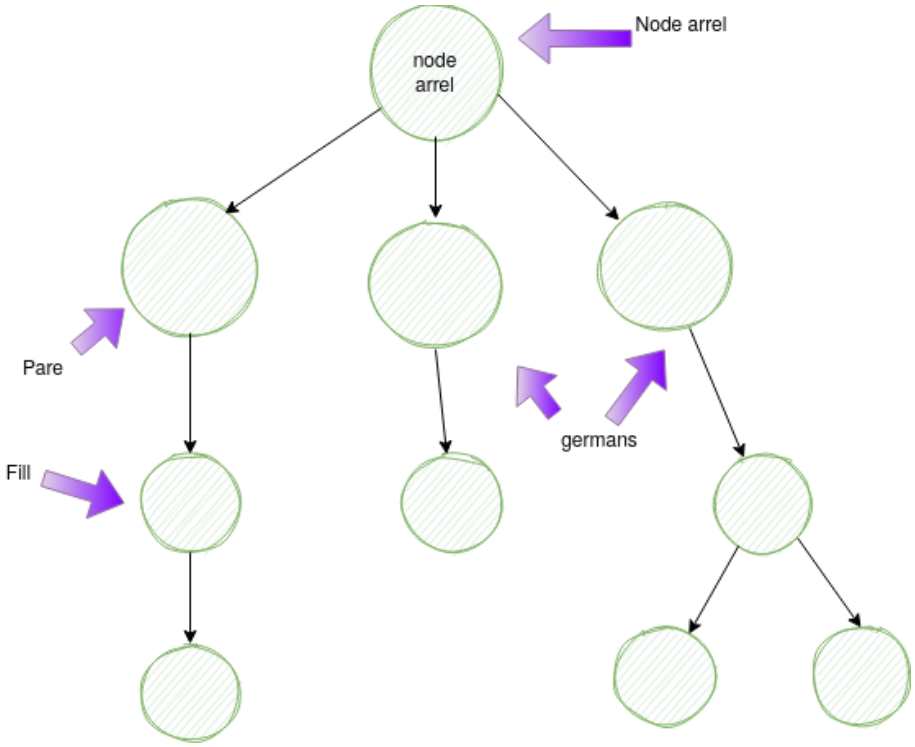

Figure 1: Estructura document XML

Per exemple, si volem tenir una agenda, crearem l'element <persona> que conté els elements <nom>, <cognoms> i telèfons que a la seva vegada pot contenir múltiples numeros.

```
<persona>
  <nom>Joan</nom>
  <cognoms>Pérez Garcia</cognoms>
  <telefons>
  <movil>6658654523<movil>
  <fixe>9375856554</fixe>
  </telefons>
<persona>
```

Activitat 3. Representa aquest esquema en format d'arbre. Respecte el node "cognoms" indica la relació que tenen amb la resta de nodes. Finalment, canvia l'estructura per afegir <telefon_feina> <mail_personal> i <mail_feina>.

Activitat 4. Fes l'esquema per emmatzemar el menu diari d'un restaurant.

2.4. Model de dades d'un document XML. Nodes.

L'estructura d'un document XML està formada pels següents tipus de nodes:

Arrel.

Aquest element s'anomena node arrel i es designa com "/". Es fa servir principalment per recòrrer l'arbre XML i processar la resta de nodes. El node arrel només hi pot haver-hi un, i no té ni ascendents ni germans, només descendents.

Elements.

És la unitat bàsica dels documents XML i contenen la informació. Delimiten la informació fent servir una etiqueta d'obertura i un altre de tancament. Entre aquestes etiquetes es troba el contingut de l'element, que pot ser una dada, altres elements o estar buit.

Figure 2: Elements XML

Elements buits.

Poden tenir atributs, però no tenen contingut i s'obre i es tanca amb una sola etiqueta.

Atributs.

Són parells nom-valor que permeten especificar dades adicionals d'un element. Estan ubicats a l'etiqueta d'obertura. Es fan servir per emmagatzemar informació adicional sobre el contingut. Per exemple, per especificar les unitats de mesura:

```
<pes unitats="Kg"> 80 </pes>
```

També es poden fer servir per indentificar un element per distingir-lo d'un altre:

```
<capitulo materia="Lenguaje Marcas">
  <tema> Introducción </tema>
  <capitulo_1 subapartado="1"> Conceptos y ventajas </capitulo_1>
  <capitulo_1 subapartado="2"> SGML el origen </capitulo_1>
</capitulo>
```

Texte.

El texte representa les dades d'un document XML. Pot aparèixer com a contingut d'un element o com a valor d'un atribut, i no pot aparèixer en cap altre lloc. Per a representar els espais en blanc tenum quatre tipus de caràcters a XML

Nom	Simbol	Codi	Abreviatura
Tabulador	\t		TAB
Nova línea	\n
LF
Retorn de carro	\r
CR
Espai	\s		SPACE

Els espais en blanc entre elements són ignorats.

Comentaris.

Els comentaris comencen pels caràcters <!– i es tanquen amb els caràcters –>. Dintre dels comentaris es pot escriure qualsevol caràcter excepte el doble guió (–) que pot confondre l'analitzador amb una etiqueta de tancament. Els comentaris, com en el cas de la programació, són fonamentals per a la comprensió del document XML, ja que és on explicarem per a que serveix cadascun dels elements. Els comentaris són molt últils quan tornem al nostre document al cap d'un temps o quan l'utilitzen altres programadors.

Instruccions de processament.

Comencen per <? i terminen per ?>. Són instruccions per al processador XML (el programa que analitza els documents XML) de manera que aquestes instruccions són independents del document XML. No formen part del document XML. Es fan servir per especificar la versió de XML que fem servir, la codificació del texte i per últim indiquem si el fitxer té altres fitxers externs de definició associants.

Exemple:

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

Entitats predefinides.

Són entitats que permeten escriure alguns caràcters especials de marcatge, i per tant són interpretades. A la següent taula podem veure les entitats predefinides a XML.

Entitat	Caràcter
&amp;	&
&lt;	<
&gt;	>
&apos;	'
&quot;	"

Seccions CDATA.

Són equivalents als comentaris, donat que l'analitzador (parser) XML no els processarà. No poden aparèixer ni abans de l'element arrel ni després del seu tancament. Fa servir l'etiqueta <[CDATA[per l'inici i]]> per indicar la fi.

Definicions de tipus de document (DTD). Els documents DTD i XSD

Permeten definir regles que imposen restriccions sobre els elements XML. La seva existència no és obligatoria per tal que el document estigui ben format, però si hem d'afegir aquesta línea si volem tenir un document vàlid.

Exemple complet:

2.5. Editors XML

Existeixen moltes eines per a la gestió d'XML i les seves tecnologies associades. Algunes són molt sofisticades i en general amb llicències propietàries, però també tenimi més senzilles, de codi obert i gratuïtes.

Alguns exemples d'aplicacions comercials:

XML Spy

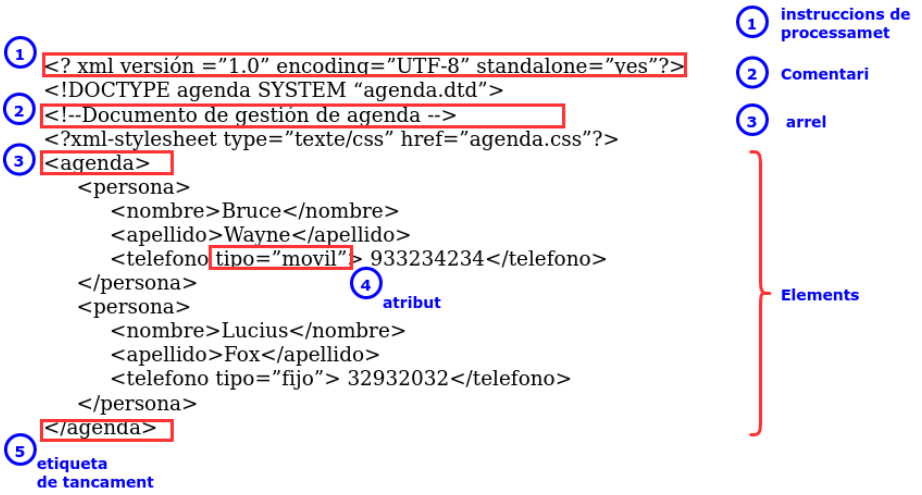

Figure 3: Exemple XML

OXygen

Per altra banda tenim el següent software Open Source:

XML Copy Editor (multiplataforma)

XML Notepad

Alguns editors de propòsit general també poden configurar-se per poder analitzar arxius XML. Alguns dels més coneguts són:

Visual Studio Code RECOMANAT

Activitat 5. Instal.lar i configurar la nostra eina d'edició d'XML i provar el nostre primer codi XML en l'editor.

2.5. Elements. Regles i consideracions

Per definir els noms dels elements també s'han definit una sèrie de regles molt bàsiques:

- Els elements poden començar amb una lletra (de l'alfabet de qualsevol idioma), subratllat o dos punts (estan desaconsellats perquè es fan servir pels espais de noms, que veurem a continuació).
- Els següents caràcters poden ser lletres, dígits, subratllats, guió baix, coma o dos punts (a-zA-Z / 0-9 / _ / , / ;).
- Els noms que comencen per XML estan reservats per a l'estandar.
- No pot haver-hi espais en blanc, ni cap altre caràcter no mencionat anteriorment: $, @, etc–

Activitat 6. Omple la següent taula:

Nom	Correcte Si/No	Motiu
<Nombre_Persona> </Nombre_persona> <1_Codigo> </1_Codigo> <día/mes/año> </día/mes/año> <Delegado provincial> </Delegado provincial> <Código-Interno> </Código-interno>		

Elements vs Atributs Les dades poden ser emmagatzemades tant als elements com als atributs. Per exemple, podem veure els següents exemples:

Ús d'atributs:

```
<person sex="female">
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

La mateixa informació fent servir elements:

```
<person>
  <sex>female</sex>
  <firstname>Anna</firstname>
  <lastname>Smith</lastname>
</person>
```

No hi ha regles per definir quan s'ha d'utilitzar un element o un atribut. No obstant, podem donar alguns consells generals.

Elements:

- Representen jerarquíes o continguts uns dintre dels altres.
- Hi poden haver altres elements dintre d'un element.
- L'ordre dels elements es representatiu.
- Poden tenir atributs.
- Poden haver múltiples ocurrències del mateix element dintre del document.

És a dir, poden haver diversos elements amb el mateix nom.

Atributs:

- Van associats als elements. Pot haver-hi més d'un per a cada element.
- Són modificadors de la informació.
- Es solen fer servir per a registrar metadades.
- L'ordre dels atributs dintre de l'element no és rellevant.
- No es poden extendre amb altres elements al seu interior.
- Un atribut no pot tenir diferents valors dintre d'un mateix element.
- Els atributs es fan servir sovint a HTML, però a XML en general s'ha d'intentar evitar-los.

2.6. Analitzadors XML (XML Parser)

Els parser analitzen els documents XML i determinen l'estructura i propietats de les dades. Alguns parsers generen l'arbre asociat a l'arxiu i ens ho mostren pel navegador. Els parsers verifiquen la sintaxi del document. Si l'analitzador també es verificador, aleshores és un analitzador verificador. Aquests analitzador també comproben la semàntica del document e informen dels error existents. Existeixen validadors en línia com XML Validation http://www.xmlvalidation.com/.

Activitat 7. Valida alguns dels exemples anteriors amb el validador online.

2.7. Espai de noms

Els noms dels elements XML els escull el desenvolupador. Podem tenir diferents aplicacions que generin elements amb idèntic nom però diferent significat. Aquí tenim un codi XML que fa servir l'element title per descriure el títol d'una entrada bibliogràfica:

```
<book>
  <title>A true story</title>
  <description>A real cool publication</description>
</book>
```

A continuació tenim un segon fragment de XML on l'element title descriu el títol d'una persona.

```
<record>
  <name>Miller</name>
  <title>Dr. </title>
  <publications> ... </publications>
</record>
```

Si unim aquests dos fragments, hauria un conflicte de noms perquè els dos fragments contenen l'element <title> amb un significat diferent (títol d'un llibre I títol d'una persona). Per resoldre aquest problema XML fa servir els espais de noms o namespaces.

Per utilitzar els espais de noms primer hem de definir-los. Per declarar un espai de noms podem afegir l'atribut xmlns a l'element d'obertura, amb la següent sintaxi:

```
<element xmlns:prefix="URI">
```

Per exemple:

```
<b:book xmlns:b="https://martinfowler.com/articles/writingInXml.html">
  <b:title>A true story</b:title>
  <b:description>A real cool publication</b:description>
</b:book>
```

A tenir en compte:

- La URI no ha d'apuntar a cap web que estigui on-line, és només un identificador únic. Si volguèssim desenvolupar un llenguatge XML per un àmbit, hauriem de tenir una web on hi hagués un índex de tots els elements que es poden fer servir.
- Quan fem servir un element d'aquell espai de noms, afegim el prefix seguit del nom de l'element, separat per dos punts.

```
<prefix:element>data</prefix:element>
```

L'inconvenient d'aquest sistema és que hem d'afegir el prefixe per a cada element. Si fem servir majoritàriament un espai de noms, podem definir-ho com a espai de noms per defecte. Simplement treiem el prefix:

```
<element xmlns="URI">
```

En aquest cas, ja no haurem d'afegir el prefix a cada element. Per exemple, per escriure un doment en XHTML:

```
<?xml version="1.0"?>
<!-- elements are in the HTML namespace by default -->
<html xmlns='http://www.w3.org/1999/xhtml'>
  <head>
    <title>Frobnostication</title>
  </head>
  <body>
    <p>Moved to <a href='http://frob.example.com'>here</a>.</p>
  </body>
</html>
```

Els espais de noms s'utilitzen en la validació de documents XML (XML Schema / XSD).

Referències:

Edutech Wiki XML namespace

2.8. Correció sintàctica a XML

L'especificació XML defineix la sintaxi que el llenguatge ha de seguir:

- Com es delimiten els elements amb etiquetes
- Que format pot tenir una etiqueta
- Que noms són acceptables per a elements
- On es col.loquen els atributs

Un document XML es diu que està ben format quan segueix les regles establertes per W3C per a les especificacions. Algunes ja les hem vistes:

- Només pot haver-hi un element arrel.
- Els elements que no són buits han de tenir una etiqueta d'obertura i una de tancament.

- Els element han d'aparèixer correctament eniuats, quant a la seva obertura i tancament. No poden encavalcar-se.
- Els noms dels elements i atributs són sensibles a majúscules i minúscules (case sensitive).
- Els valors dels atributs han d'aparèixer entre cometes simples o dobles pero no s'han de mesclar.
- No pot haver-hi dos atrinuts amb el mateix nom associat al mateix element.
- No es poden introduir instruccions de procesament ni comentaris en cap lloc de l'interior de les etiquetes d'obertura o tancament dels elements.
- No pot haver res iabans de l'instrucció de procesament.

Consells de disseny

1. Nivell desglossament dades

Fins a quan hem de desglossar les nostres dades, partint-les en unitats més petites?

Els valors de les dades s'han de desglossar al nivell més baix possible. Això els permet ser processats de diverses formes per diferents usos, com a visualització, operacions matemàtiques i validació de les dades. És molt més fàcil concatenar dos valors de dades de nou que no pas dividir-los. A més, les dades més granulars són molt més fàcils de validar.

És una pràctica força habitual posar un valor de dades i les seves unitats en el mateix element, per exemple <length>3cm</length>. Tanmateix, l'enfocament preferit és tenir un atribut per a les unitats, per exemple <length units="cm">3</length>.

L'ús d'un únic valor concatenat és limitant perquè:

- És extremadament complicat de validar. S'ha d'aplicar un patró complicat que hauria de canviar cada vegada que s'afegeix un tipus d'unitat.
- No es poden realitzar comparacions, conversions ni operacions matemàtiques amb les dades sense dividir-les.
- Si es vol mostrar l'element de dades de manera diferent -per exemple, com "3 centímetres" o "3 cm" o només "3"-, l'heu de dividir. Això complica els fulls d'estil i les aplicacions que processen el document d'instància.

També és possible anar massa lluny. Per exemple, podeu desglossar una data de la manera següent:

```
<dataComanda>
  <any>2001</any>
  <mes>06</mes>
  <dia>15</dia>
</dataComanda>
```

Probablement això sigui excessiu, tret que tingueu una necessitat especial de processar aquests elements per separat.