

# Llenguatge de marques i sistemes de gestió de la informació

R.A 1 : Reconeix les característiques de llenguatges de marques analitzant  
i interpretant fragments de codi

# XML

**XML** són les sigles d'*extensible markup language* (llenguatge d'etiquetatge extensible). És un llenguatge estàndard, una recomanació del World Wide Web Consortium (W3C, [www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml)). Està basat en l'estàndard ISO SGML.

En realitat l'XML no és un llenguatge de marques, sinó que és un llenguatge que ens permetrà definir els nostres llenguatges de marques propis. Això vol dir que podrem definir llenguatges de marques específics per a cada un dels camps d'interès. Si treballem en el món dels gràfics podem definir el nostre llenguatge específic per definir aquests gràfics, i si treballem en el món de la premsa podrem definir el nostre llenguatge per representar les notícies.

Per aquest motiu sovint es diu que l'XML és un **metallenguatge**, ja que ens permet definir l'estructura i el vocabulari d'altres llenguatges de marques.

Principals característiques:

- Defineix la sintaxi genèrica per marcar les dades amb valors comprensibles per als humans.
- És una manera de donar format als documents que és prou flexible per ser personalitzada per a diferents destinacions: web, impressores, bases de dades, etc.
- Està pensat perquè tothom el pugui fer servir sigui quina sigui la seva àrea d'interès.

# XML - Elements

La base de l'XML són els elements. Un element normalment estarà format per l'obertura d'una etiqueta –amb atributs o sense–, un contingut –que també pot ser un grup d'etiquetes–, i el tancament de l'etiqueta.

```
<nom>Pere Martí</nom>
```

Es podria fer l'exemple una mica més complex afegint-hi un atribut.

```
<nom càrrec="director">Pere Martí</nom>
```

Un altre dels aspectes bàsics de l'XML és que no es preocupa de la presentació sinó que parteix de la idea que l'important és el contingut de les dades i no la manera com es visualitzaran.

**L'XML permet separar el contingut de la manera com serà presentat aquest contingut als usuaris.**

# XML - Etiquetes

Les etiquetes es defineixen dins del document XML i es defineix un format per separar-les clarament del contingut de dades. Estrictament parlant hi ha dos tipus d'etiquetes:

- **Les etiquetes d'obertura** es defineixen amb els símbols de més petit "<" i més gran ">" amb un nom d'etiqueta al mig.
- **Les etiquetes de tancament** es defineixen igual que les d'obertura però a l'hora de començar l'etiqueta s'hi especifiquen dos símbols: un símbol de més petit i una barra "</".

```
<etiqueta>Contingut</etiqueta>
```

**Les etiquetes que s'obrin sempre s'hauran de tancar.**

L'XML no defineix cap etiqueta sinó que permet que les defineixi cadascú en funció del que necessiti representar. Això fa més senzill crear documents XML, ja que no cal conèixer cap etiqueta per poder començar a treballar i perquè permet adaptar-se a qualsevol tasca.

L'especificació XML defineix clarament com s'han de crear les etiquetes en els documents XML, però en canvi no defineix cap tipus ni significat associat a cap de les etiquetes. Si, per exemple, fem servir en HTML l'etiqueta `<b>`, aquesta ens està indicant que el contingut que contindrà s'haurà de representar en negreta. Això obliga que per fer documents XML s'hagin de conèixer les etiquetes. Tot i això per un motiu de llegibilitat i d'interpretació de les dades **es recomana que les etiquetes siguin autoexplicatives i pronunciables.**

# XML - Contingut

El contingut d'un element serà tot allò que hi hagi entre les etiquetes d'obertura i de tancament. En el contingut podem tenir simplement text com aquí:

```
<persona>Pere Martí</persona>
```

O bé el contingut poden ser altres elements. En l'exemple següent l'element `<persona>` conté dos elements més: `<nom>` i `<cognom>`, que a més tenen contingut dins seu:

```
<persona>  
  <nom>Pere</nom>  
  <cognom>Martí</cognom>  
</persona>
```

Una tercera possibilitat seria combinar els dos casos anteriors i que el contingut sigui una mescla de text i elements.

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

També és possible definir etiquetes sense contingut de manera que el seu significat està determinat pel nom de l'etiqueta.

# XML - Contingut - Entitats

L'única cosa que s'ha de tenir en compte a l'hora de definir continguts és que hi ha una sèrie de caràcters que poden provocar que hi hagi confusions a l'hora d'interpretar el document XML, i per tant s'han declarat il·legals. No cal oblidar que un dels objectius de l'especificació és “que sigui fàcil escriure programes que processin els documents”, i per tant s'han d'evitar les interpretacions.

Si passem el codi següent a un programa d'ordinador tindrem un problema:

```
<algebra>  
x<y i y>z  
</algebra>
```

Per a un humà això no seria un problema perquè pot interpretar les dades per mitjà del seu coneixement, però en canvi un programa en llegir el contingut del document interpretaria que `<y i y>` és el començament d'una nova etiqueta.

Per evitar aquest problema s'han definit una sèrie de valors, anomenats **entitats**, que es fan servir per poder incloure els caràcters problemàtics dins del contingut de les etiquetes

Símbol	Substitució
<	&lt;
>	&gt;
"	&quot;
'	&apos;
&	&amp;

# XML - Contingut - CDATA

XML permet fer servir les seccions CDATA dins del contingut d'un element, en el cas que el contingut d'un element sigui HTML o bé codi en algun llenguatge de programació i pugui ser interpretat.

Les seccions CDATA funcionen com les etiquetes normals. Es defineixen començant per **<![CDATA[** abans d'especificar el contingut i s'acaben amb la combinació de caràcters **]]>**. Per tant, podem definir contingut amb símbols prohibits dins de les seccions CDATA sense problemes.

```
<valor>
  <![CDATA[
    if (x <y and y>z)
    {
      printf("Correcte!");
    }
  ]]>
</valor>
```

Un altre exemple:

```
<titol>Els documents HTML comencen per &lt;html&gt;</titol>
```

```
<titol>
  <![CDATA[
    Els documents HTML comencen per <HTML>
  ]]>
</titol>
```

# XML - Atributs

Els atributs són un parell de valors separats per un = que **només s'especifiquen en les etiquetes d'obertura**. El primer valor del parell ens indica quin és el nom del contingut i es farà servir per interpretar què indiquen les dades que té associades, mentre que el segon ens definirà el contingut de l'atribut.

```
<nom carrec="professor">Frederic Garcia</nom>
```

Es pot veure com l'atribut `carrec` dona un nivell més d'informació a `nom`. Ara se sap que fa referència al nom d'un professor.

Per especificar els atributs es poden fer servir tant les cometes dobles com les cometes simples. Això sí, sempre s'ha de tancar les cometes tal com s'han obert. `<corredor posicio="1">` o `<corredor posicio='1'>` però mai `<corredor posicio="1'>`

Fins i tot quan els atributs no cal que tinguin cap valor, perquè es considera que ja se'n pot interpretar el significat pel nom de l'atribut, s'hi ha d'especificar valor entre cometes.

```
<alumne delegat="">Jaume Ravent</alumne>
```

L'XML **permet definir tants atributs com faci falta** sense cap tipus de restricció. Les úniques condicions que hem de seguir és que cada un dels atributs ha d'estar separat dels altres almenys per un espai. **L'ordre en què apareixen els atributs no té cap importància. És important remarcar que o es poden repetir els noms dels atributs dins d'un mateix element.**

```
<persona nom="Xavier" cognom="Sala" />
```

```
<persona cognom="Sala" nom="Xavier" />
```



# XML - Atribut xml:lang

L'atribut `xml:lang` és un atribut predefinit i serveix per especificar l'idioma que s'ha fet servir per escriure el contingut tant de l'element com dels altres atributs. El valor de l'atribut `xml:lang` pot ser buit o bé ha d'estar en una de les formes definides per l'IETF en el BCP 47

```
<missatge xml:lang="ca">Hola</missatge>
```

Però també seria vàlid fer servir qualsevol de les combinacions de regió i idioma:

```
<missatge xml:lang="ca-ES">Hola</missatge>  
<missatge xml:lang="ca-AD">Salut</missatge>
```

Si mirem el següent cas:

```
<saludar xml:lang="ca">  
  <arribar>Hola</arribar>  
  <marxar>Adéu</marxar>  
</saludar>
```

`arribar` i `marxar` tenen també l'atribut `xml:lang` per què són contingut de `<saludar>`, i per tant el seu contingut i el valor dels seus atributs està en català.

Sempre es pot evitar aquesta forma d'herència de l'atribut redefinint l'atribut `xml:lang` com en aquest exemple:

```
<saludar xml:lang="ca">  
  <arribar>Hola</arribar>  
  <arribar xml:lang="en">Hello</arribar>  
</saludar>
```

# XML - Atribut xml:space

L'atribut `xml:space` fa referència a com s'han de tractar els espais que hi ha en el contingut d'un element determinat.

Quan algú edita un document XML és corrent que faci servir espais i salts de línia perquè el document sigui més “bonic” però realment aquests espais i salts de línia no formen part del contingut. **Amb l'ús de l'atribut `xml:space` es defineix si aquests espais són part del contingut o no.**

Valors acceptats:

Valor	Significat
<b>default</b>	Implica que el document ha de mostrar només un sol espai de separació entre les paraules. Qualsevol altra cosa ha de ser suprimida.
<b>pre-serve</b>	Fa que els espais es deixin tal com estan en el document. Per tant, si hi ha 5 espais després d'una paraula en el contingut s'han de preservar aquests espais.

Exemple:

```
<missatge xml:space="default">
Hola:
  Com va tot?
</missatge>
```

al ser default obtindrem :

```
<missatge>Hola: Com va tot?</missatge>
```

Si hagués sigut preserve s'hagués mostrat igual com està en l'exemple

```
<missatge>
Hola:
  Com va tot?
</missatge>
```

# Noms vàlids XML

Les regles per definir noms correctes són senzilles:

1. Els noms han de començar per una lletra de l'alfabet, el caràcter de subratllat (\_) o un guió (-). També s'accepta el caràcter de dos punts (:), però està reservat.
2. Els caràcters en majúscules són diferents dels caràcters en minúscules.
3. No hi pot haver espais enmig del nom.
4. No poden començar per la paraula *XML* tant si qualsevol de les lletres està en majúscules o en minúscules. Aquestes paraules es reserven per a estandarditzacions futures.

Dos exemples **incorrectes** que no segueixen aquestes regles: (sabries dir perquè?)

```
<Alt Empordà></Alt Empordà>
```

```
<carrera>  
  <1aPosicio>Manel Garcia</1aPosicio>  
  <2aPosicio>Pere Pi</2aPosicio>  
</carrera>
```

# Comentaris XML

Sovint en els documents XML s'especifiquen dades extra que realment no formen part del document. Aquestes dades s'anomenen *comentaris* i es fan servir per a moltes coses diverses, com ara:

- Generar documentació sobre el document.
- Indicar a la gent que pugui rebre el document què es volia fer en crear-lo.
- Altres.

Els comentaris s'especifiquen incloent-los entre els símbols `<!--` i `-->` i es poden posar comentaris en qualsevol lloc del document XML excepte dins de les etiquetes. Com a exemple, en el document següent s'ha inclòs el comentari “Llista d'alumnes” per a indicar en quin lloc comença la llista d'alumnes:

```
<professors>
  <nom>Marcel Alaba</nom>
</professors>
<!-- Llista d'alumnes -->
<alumnes>
  <nom>Frederic Garcia</nom>
  <nom>Federicu Pi</nom>
</alumnes>
```

# Instruccions del procés

Les **instruccions de procés** són una manera de donar instruccions als programes que llegiran el document. Es defineixen dins dels símbols "<?" i posteriorment **sempre s'hi ha d'especificar a quin programa van dirigides** amb l'única restricció que no poden ser les lletres "XML" en qualsevol combinació de majúscules o minúscules.

Per tant, per afegir informació que vagi destinada a un programa anomenat **php** s'han de definir les instruccions de procés amb una instrucció de procés:

```
<?php ... ?>
```

Aquest sistema permet que en un document XML es puguin afegir instruccions de procés diferents per a programes diferents:

```
<?programa1 ... ?>  
<?programa2 ... ?>
```

El contingut de les dades pot ser qualsevol cosa que tingui sentit per al programa que les llegirà, i per tant no necessàriament ha de tenir sentit per als processadors XML. Com que no formen part del document XML en si poden aparèixer en qualsevol lloc del document excepte en mig d'una etiqueta.

# Declaració XML

L'especificació és una línia que defineix una manera d'identificar que un document és XML per mitjà d'una etiqueta especial anomenada *declaració XML*, que serveix per indicar que un document és XML.

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

**La declaració XML és opcional, tot i que es considera recomanable**, ja que té alguns atributs que poden ajudar als programes a entendre característiques del document com el codi de caràcters que s'hi fa servir, la versió d'XML que s'ha usat, etc.

Atributs:

Atribut	Objectiu	Obligatori ?
<b>version</b>	Defineix quina versió d'XML s'ha fet servir per crear el document.	Sí
<b>encoding</b>	Defineix el codi de caràcters que fa servir el document.	No
<b>standalone</b>	Serveix per indicar si el document no depèn d'altres documents.	No

# Declaració XML - Atribut encoding

L'atribut **encoding** permet definir amb quin codi de caràcters s'han codificat les dades.

Per defecte tots els programes que llegeixin XML han de poder interpretar els codis de caràcters que estiguin en UTF-8 i UTF-16 i fins i tot detectar en quina de les dues codificacions està fet el document. Això implica que si el document està creat amb alguna d'aquestes codificacions no caldria especificar l'atribut **encoding**. Tot i això, per motius de llegibilitat del document sovint s'hi especifica igualment.

Com que l'estàndard ASCII és un subconjunt d'UTF-8 els documents que estiguin en ASCII tampoc no requereixen cap declaració.

Tipus	Codis de caràcters
Unicode	UTF-8, UTF-16, ISO-10646-UCS-2, i ISO-10646-UCS-4
ASCII i variants	ISO-8859-1, ISO-8859-2, ..., ISO-8859-n
JIS X-0208-1997	ISO-2022-JP, Shift_JIS, EUC-JP

# Declaració XML - Atribut standalone

Si un programa tracta aquest document ha de tenir en compte aquest valor, i per tant haurà de llegir el document que declara els atributs per defecte.

Amb l'atribut **standalone** s'està definint si el document XML es pot entendre per si sol o bé necessita que sigui interpretat amb l'ajuda d'algun altre document, i per aquest motiu només té dos valors possibles

Valor	Significat
yes	El document és complet, i per tant no li calen altres documents per ser interpretat.
no	El document no es pot entendre per si sol i n'hem de llegir d'altres per poder-lo entendre.



# Normes XML

En general podem definir aquestes regles com:

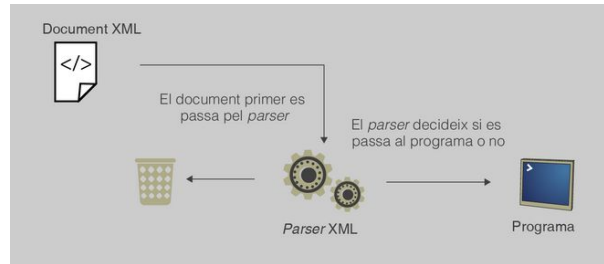
- Només hi pot haver un element arrel.
- Totes les etiquetes que s'obren s'han de tancar.
- Les etiquetes han d'estar imbricades correctament.
- Els noms de les etiquetes han de ser correctes.
- Els valors dels atributs han d'estar entre cometes.

```
<persones>
  <persona>
    <nom>Pere</nom>
    <cognom> Pérez </cognom>
  </persona>
  <persona>
    <nom>Marià</nom>
    <cognom>Garcia</cognom>
  </persona>
</persones>
```

```
<classe nom="Llenguatges de marques">
  <alumne delegat='si'>Pere Garcia</alumne>
  <alumne>Frederic Pi</alumne>
</classe>
```

# Processadors XML

L'objectiu principal de totes les regles que defineixen com s'ha d'escriure un document XML és definir una manera d'escriure documents XML que puguin ser llegits i interpretats per un programa d'ordinador. Els programes encarregats de detectar si un document XML està ben format s'anomenen genèricament **processadors d'XML**, tot i que sovint es fa servir el nom en anglès, ***parsers***.



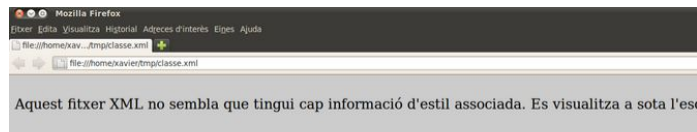
A part de definir com s'han de crear els fitxers XML l'especificació també defineix com han de reaccionar els processadors davant dels diferents errors que puguin detectar en tractar un fitxer XML. Els errors es defineixen en dos grups:

- **Errors lleus:** els errors lleus són els que incompleixen alguna de les regles definides com a *recomanacions*. Si algun document XML incompleix alguna recomanació l'analitzador pot continuar analitzant el document intentant recuperar-se de l'error.
- **Errors greus:** es generen quan s'incompleix alguna de les regles *obligatòries* de l'especificació. Es defineix que en cas de trobar algun error greu l'analitzador s'ha d'aturar immediatament i no continuar amb el procés d'anàlisi. En general qualsevol error que faci que el document no sigui "ben format" serà un error d'aquest tipus.

# XML - Navegador web

Un cop s'ha creat un document XML normalment caldrà comprovar que aquest document està **ben format** segons les regles d'XML fent servir un processador XML.

La manera més senzilla de comprovar que un document està ben format és carregar-lo des d'un navegador. La majoria dels navegadors porten un processador XML incorporat i detecten automàticament que un document és XML i el comproven



```
- <classe>
- <professor>
  <nom>Marcel</nom>
  <cognom>Puig</cognom>
</professor>
- <alumnes>
- <alumne>
  <nom>Filomeno</nom>
  <cognom>Garcia</cognom>
  <nota>6</nota>
</alumne>
- <alumne>
  <nom>Frederic</nom>
  <cognom>Pi</cognom>
  <nota>4</nota>
</alumne>
- <alumne>
  <nom>Manel</nom>
  <cognom>Puigdevall</cognom>
  <nota>9</nota>
  <delegat/>
</alumne>
</alumnes>
</classe>
```

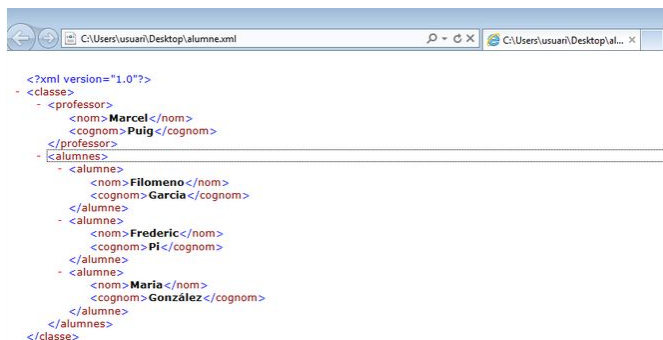
# Processador XML - MSXML

Microsoft XML Parser és la biblioteca oficial per processar XML en molts dels productes de Microsoft. Es tracta d'una biblioteca inclosa en el sistema operatiu, i per tant s'actualitza via Windows Update. N'han anat sortint versions en què s'han anat afegint funcions. Es pot fer servir des de diferents llenguatges de programació, llenguatges de *scripts* o des de llenguatges basats en .NET

Fan servir MSXML productes com Microsoft Windows, Microsoft Office, Microsoft SQL Server, Microsoft Internet Explorer...

Molts dels programes que processen XML en sistemes Windows solen fer servir aquesta biblioteca (especialment si treballen en .NET).

Com que Internet Explorer fa servir aquesta biblioteca, simplement carregant un document XML amb Internet Explorer en realitat s'està comprovant amb l'MSXML



# Estructura documents XML

Una de les coses que s'aconsegueixen en forçar que els documents XML segueixin les seves normes és fer que la informació que conté un document s'organitzi de manera **jeràrquica**.

```
<?xml versión="1.0" encoding="UTF-8" ?>
<classe>
  <professor>
    <nom>Marcel</nom>
    <cognom>Puig</cognom>
  </professor>
  <alumnes>
    <alumne>
      <nom>Filomeno</nom>
      <cognom>Garcia</cognom>
    </alumne>
    <alumne>
      <nom>Frederic</nom>
      <cognom>Pi</cognom>
    </alumne>
    <alumne>
      <nom>Manel</nom>
      <cognom>Puigdevall</cognom>
      <delegat/>
    </alumne>
  </alumnes>
</classe>
```

En aquest exemple, hi ha un element arrel anomenat **<classe>** que engloba tot el document, i com a contingut té dos elements més, **<professor>** i **<alumnes>**. Els elements que formen part del contingut d'un node s'anomenen **fills**.

Per tant, **<classe>** té dos elements **fills**: **<professor>** i **<alumnes>**. Alhora l'element **<professor>**, que és un dels fills de **<classe>**, també té dos fills: **<nom>** i **<cognom>**.

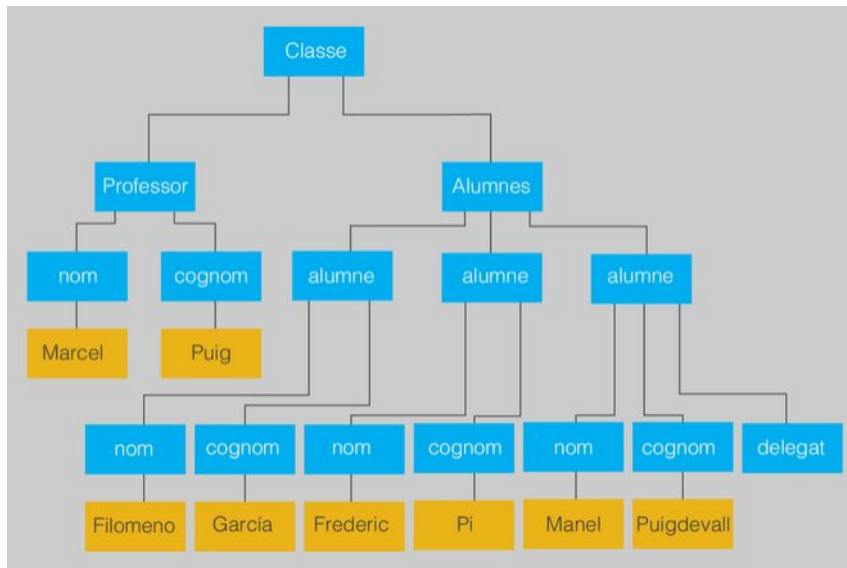
**<nom>** i **<cognom>** són fills de **<professor>** i **néts** de **classe**

Els elements **<nom>** i **<cognom>** no tenen nodes fills sinó que contenen les dades del document. En aquest cas dues cadenes de text **Marcel** i **Puig**. Els nodes finals s'anomenen **fulles** i generalment seran sempre nodes de dades.

# Representació en arbres

Una de les característiques interessants dels documents XML és que, en tenir les dades estructurades de manera jeràrquica, aquesta jerarquia fa que els documents XML puguin ser representats gràficament en forma d'arbre. La representació en arbre és útil per comprendre millor quina és l'estructura del document.

```
<?xml version="1.0" encoding="UTF-8" ?>
<classe>
  <professor>
    <nom>Marcel</nom>
    <cognom>Puig</cognom>
  </professor>
  <alumnes>
    <alumne>
      <nom>Filomeno</nom>
      <cognom>Garcia</cognom>
    </alumne>
    <alumne>
      <nom>Frederic</nom>
      <cognom>Pi</cognom>
    </alumne>
    <alumne>
      <nom>Manel</nom>
      <cognom>Puigdevall</cognom>
      <delegat/>
    </alumne>
  </alumnes>
</classe>
```



# Creació documents XML

Generalment els documents XML seran creats i llegits des de programes d'ordinador, però algunes vegades també es pot donar el cas que s'hagin de crear manualment.

Crear un document XML manualment és molt més senzill que crear un document binari, ja que no difereix gaire de crear un document de text. Simplement necessitem un editor de text que no enriqueixi el text.

L'única cosa que cal per crear un document XML és un **editor de text normal** i corrent que no enriqueixi el text. Els documents de text que permeten afegir format, com ara text en negreta, canviar el tipus de lletra, etc., com per exemple el Microsoft Word, l'OpenOffice.org Writer, el LibreOffice Writer, etc., solen generar documents de text enriquit que es concentren més en com s'ha de mostrar el document que no pas en les dades. A més, aquests programes sovint canvien automàticament alguns caràcters del text que anem escrivint (les cometes solen ser canviades sistemàticament) per fer-lo més "agradable" a l'hora d'imprimir-lo.

La realitat és que per crear documents XML van molt millor els editors més senzills (Gedit, Bloc de notes, vi, etc.) que no pas els editors de text enriquit.

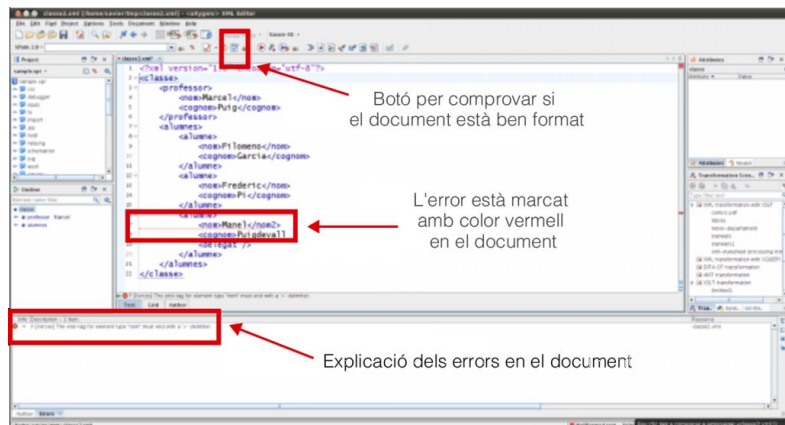
En alguns casos aquests editors més senzills fins i tot detecten que s'està editant un document XML i marquen amb colors diferents les etiquetes i les dades

```
<?xml version="1.0" ?>
<classe>
  <professor>
    <nom>Marcel</nom>
    <cognom>Puig</cognom>
  </professor>
  <alumnes>
    <alumne>
      <nom>Filomeno</nom>
      <cognom>Garcia</cognom>
      <nota>6</nota>
    </alumne>
    <alumne>
      <nom>Frederic</nom>
      <cognom>Pi</cognom>
      <nota>4</nota>
    </alumne>
    <alumne>
      <nom>Manel</nom>
      <cognom>Puigdevall</cognom>
      <nota>9</nota>
      <delegat/>
    </alumne>
  </alumnes>
</classe>
```

# Editors especialitzats en XML

Tot i que amb els altres editors es poden crear documents XML, quan es vol fer un treball professional normalment s'ha d'acabar recorrent a un editor d'XML. Aquests editors estan dissenyats específicament per crear i editar documents XML de manera eficient i senzilla minimitzant les possibilitats que es cometin errors en l'edició. Generalment tots ofereixen un entorn amb un grup de finestres amb diferents vistes de l'edició per intentar que no es perdi la visió de conjunt del que s'està creant.

Per exemple, l'editor EditiX ens va creant l'arbre XML alhora que anem escrivint el document XML



Els principals editors comercials:

**oXygen XML Editor** (Windows, Linux, Mac OS X)

**Editix XML Editor** (Windows, Linux, Mac OS X)

**Altova XMLSpy XML editor** (Windows)

**Stylus Studio** (Windows)

**XMLmind** (Windows, Linux, Mac OS X)

**XMLwriter** (Windows)

**Liquid XML Studio** (Windows)

**Serna Enterprise XML Document Editor** (Windows, Linux, Mac OS X, Solaris)



# Creació document XML

A l'hora de definir un grup de dades dins d'un document XML caldrà fer tota una sèrie de passes prèvies que permetin determinar quines són les dades que cal emmagatzemar i posteriorment definir quina és l'estructura que s'ha de donar a aquestes dades; així doncs:

1. **Determinació de les dades:** És bàsic abans de crear un document XML saber clarament quines són les dades que s'hi han de posar. Si no s'està restringit per un programa que ja determini l'estructura del document XML, el que cal és determinar quines dades s'han d'emmagatzemar. Exemple: Volem crear una biblioteca en què es puguin emmagatzemar les dades dels llibres que hi ha. Per tant, fem una llista amb les dades que considerem que cal que hi hagi en el document.

- Títol
- Subtítol
- Autor
- Any de publicació
- Editorial
- Nom de la col·lecció
- Idioma

# Creació document XML

2. **Determinació de l'estructura:** Una altra de les coses bàsiques a l'hora de crear un document XML és definir quina és l'estructura que hauran de tenir les dades. Aquesta estructura estarà determinada per les necessitats del programa o de la persona que farà servir el document XML. De manera que les possibilitats a l'hora de crear una estructura per a les dades que volem fer servir són moltes: agrupar per autor, agrupar per llibre, agrupar per editorial, agrupar per tema... En una biblioteca podem fer l'estructura des de molts punts de vista. Per exemple, la podem fer a partir dels autors, com aquesta:

- autor 1
  - llibre 1
  - llibre 2
- autor 2
  - llibre 1
- etc.

- llibre 1
  - autor 1
- llibre 2
  - autor 1
- etc.

Fent això obtindrem un codi XML com el següent:

```
<biblioteca>
  <autor>
    <nom>John Ronald Reuel Tolkien</nom>
    <llibres>
      <llibre>
        <titol>El Hobbit</titol>
        <any>2010</any>
        <idioma>català</idioma>
        <editorial>
          <nom>Edicions de la magrana</nom>
          <col·lecció>L'Esparver</col·lecció>
        </editorial>
      </llibre>
      <llibre>
        <titol>El senyor dels anells</titol>
        <subtitol>La germandat de l'anell</subtitol>
        <any>2002</any>
        <idioma>català</idioma>
        <editorial>
          <nom>Editorial Vicens Vives</nom>
        </editorial>
      </llibre>
    </llibres>
  </autor>
  <autor>
    <nom>Isaac Asimov</nom>
    <llibres>
      <llibre>
        <titol>Jo, robot</titol>
        <any>2001</any>
        <idioma>català</idioma>
        <editorial>
          <nom>Edicions Proa</nom>
          <col·lecció>Proa Butxaca</col·lecció>
        </editorial>
      </llibre>
    </llibres>
  </autor>
</biblioteca>
```

# Espais de noms (I)

XML permet definir etiquetes personalitzades, però això pot causar conflictes quan es combinen documents de diferents fonts.

```
<biblioteca>
  <llibre>
    <titol>El Quixot</titol>
    <autor>Miguel de Cervantes</autor>
  </llibre>
  <autor>
    <nom>Gabriel García Márquez</nom>
    <nacionalitat>Colombià</nacionalitat>
  </autor>
</biblioteca>
```

És **<autor>** un autor de llibre o una definició d'autor?

# Espais de noms (II)

Solució: **Espais de Noms (Namespaces):**

- Fer úniques les etiquetes afegint prefixos basats en URLs.
- Ús de l'atribut `xmlns` per definir aquests prefixos.

```
<biblioteca xmlns:llib="http://www.exemple.org/llibres" xmlns:aut="http://www.exemple.org/autors">
  <llib:llibre>
    <llib:titol>El Quixot</llib:titol>
    <llib:autor>Miguel de Cervantes</llib:autor>
  </llib:llibre>
  <aut:autor>
    <aut:nom>Gabriel García Márquez</aut:nom>
    <aut:nacionalitat>Colombià</aut:nacionalitat>
  </aut:autor>
</biblioteca>
```

# Espais de noms (III)

Podem simplificar la solució utilitzant àlies i noms per defecte

```
<biblioteca xmlns="http://www.exemple.org/l1bres" xmlns:aut="http://www.exemple.org/autors">
  <l1bre>
    <titol>El Quixot</titol>
    <autor>Miguel de Cervantes</autor>
  </l1bre>
  <aut:autor>
    <aut:nom>Gabriel García Márquez</aut:nom>
    <aut:nacionalitat>Colombià</aut:nacionalitat>
  </aut:autor>
</biblioteca>
```

On podem veure que:

- Aquí, `<l1bre>` i els seus fills pertanyen a l'espai de noms per defecte <http://www.exemple.org/l1bres>.
- `<aut:autor>` i els seus fills pertanyen a l'espai de noms <http://www.exemple.org/autors>.