

# Unitat 3. Definició de vocabularis i esquemes en XML (PART A)



**LLENGUATGES DE MARQUES I SISTEMES DE  
GESTIÓ DE LA INFORMACIÓ  
(1º DAM)**

*Professora: Isabel Soriano*

# CONTINGUTS



## 1. INTRODUCCIÓ.

## 2. SOFTWARE XML.

## 3. DEFINICIÓ DE LA ESTRUCTURA I SINTAXI D'UN DOCUMENT XML.

### 3.1 PRÒLEG

### 3.2 COS

#### 3.2.1 ETIQUETES

#### 3.2.2 ELEMENTS

#### 3.2.3 ATRIBUTS

#### 3.2.4 COMENTARIS

#### 3.2.5 SECCIÓ 'CDATA'

## 4. ELABORACIÓ D'UN DOCUMENTO XML BEN FORMAT

# 1.Introducció



**XML (eXtensible Markup Language)** →serveix com a format d'emmagatzematge de dades i intercanvi d'informació.

**Veurem en aquest tema (tant en la part A com la part B):**

- ☐ Estructura i sintaxi de XML.
- ☐ Verificarem si estan escrits correctament i si són vàlids
- ☐ L'estructura d'un DTD i un XSD per a definir les restriccions, els tipus dels atributs i elements, etc. d'un document XML.

Les marques d'un document HTML no aporten informació rellevant als usuaris. Les de XML transmeten el significat del contingut que s'emmagatzema →crear vocabulari

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <fechaNac>2018-10-25</fechaNac>
  <DNI>fran@librolm.com</DNI>
  <nombre>Francisco Jesús González</nombre>
</usuario>
```

## 2. Software XML



- **XML Copy Editor** —> aplicació d'escriptori.
- Altres aplicacions: gedit, Brackets, etc.

### **Online**

- Code Beautify —> <https://codebeautify.org/xmlviewer>

### **VALIDACIÓ (Per a la teoria de la part B del tema)**

- **XML Copy Editor**

### **Online**

- **XML Schema generator** (DTD I XSD): <https://xml.mherman.org/>

### **Liquid Studio XML:**

- XML Validator (XSD): <https://www.liquid-technologies.com/online-xsd-validator>
- XML to XSD: <https://www.liquid-technologies.com/online-xml-to-xsd-converter>
- XSD to XML: <https://www.liquid-technologies.com/online-xsd-to-xml-converter>

### 3. Definició de la estructura i sintaxi d'un document XML



- Per a poder visualitzar-se un document XML pot fer-se directament sobre un navegador (Google Chrome, etc.).
- Un document XML s'ha de crear en text pla.
- L'estructura és jeràrquica arborescent. Conté un pare (arrel) únic i una sèrie de fills (elements secundaris). L'arrel sempre es col·loca en la part superior.
- La extensió es .xml

```
<?xml version="1.0" encoding="utf-8"?>
<padre>
  <hijo>
    <subhijos> ... </subhijos>
  </hijo>
</padre>
```

En tot document XML es distingeixen 2 parts:

- **Pròleg** → conté informació respecte al document creat (versió, codificació de caràcters, descripció d'estructura, etc.)
- **Cos** → ha de ser únic i sobre ell estan continguts els altres elements.

# 3.1 Pròleg



## **Declaració XML** (*no és estrictament una instrucció de processament*) – **NO obligatòria**

Com hem vist és la primera línia del document i inclou el següent:

- **versió** → versió XML usada (version="1.0") - obligatori
- **encoding** → informa de la codificació del document (encoding="UTF-8")
- **standalone** → informa si el document porta associat un fitxer DTD o XSD. Si porta cap associat d'aquests fitxers de manera externa (yes) si és de manera interna (no). (standalone="yes")

**Gedit**

```
<?xml version="1.0" encoding="UTF-8"?>
<arbol>
  <especie>
    <nombre>Litchi</nombre>
    <variedad>Kway May Pink</variedad>
    <origen>Filipinas</origen>
    <color>rojo</color>
    <maduración>agosto</maduración>
    <riego>diario</riego>
    <precio>3.5</precio>
  </especie>
</arbol>
```

**Google  
Chrome**

This XML file does not appear to have any style information

```
▼ <arbol>
  ▼ <especie>
    <nombre>Litchi</nombre>
    <variedad>Kway May Pink</variedad>
    <origen>Filipinas</origen>
    <color>rojo</color>
    <maduración>agosto</maduración>
    <riego>diario</riego>
    <precio>3.5</precio>
  </especie>
</arbol>
```

# 3.1 Pròleg



**Prova que genera un arbre en la següent eina en línia:**

<https://codebeautify.org/xmlviewer>

The screenshot shows the Code Beautify XML Viewer interface. On the left, the 'XML Input' section contains the following XML code:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <arbol>
3   <especie>
4     <nombre>Litchi</nombre>
5     <variedad>Kway May Pink</variedad>
6     <origen>Filipinas</origen>
7     <color>rojo</color>
8     <maduración>agosto</maduración>
9     <riego>diario</riego>
10    <precio>3.5</precio>
11  </especie>
12 </arbol>
13
```

In the center, there are buttons for 'Load Url', 'Browse', 'Tree View', 'Beautify / Format', and 'Minify'. Below these is an advertisement for Google with a 'Stop seeing this ad' button.

On the right, the 'Result : XML Tree View' section displays the XML structure as a tree:

- arbol
  - especie
    - nombre: Litchi
    - variedad: Kway May Pink
    - origen: Filipinas
    - color: rojo
    - maduración: agosto
    - riego: diario
    - precio: 3.5

**EXERCICI PROPOSAT 1– Afegir dues espècies més a l'exemple del document XML anterior.**

## 3.1 Pròleg



### Instruccions de processament

En un document XML, una instrucció de processament serveix per a indicar una certa informació al programa que processe aquest document. Les instruccions de procés s'escriuen començant amb la parella de caràcters "<?" i finalitzant amb ">".

Per exemple es podria afegir una línia en el XML per a associar un arxiu CSS:

`<?xml-stylesheet type="text/css" href="estilo-animales.css"?>`

### EXERCICI PROPOSAT 2

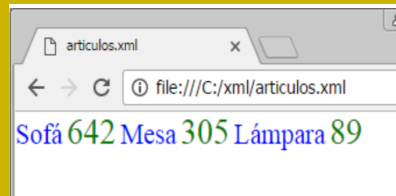
Donat l'arxiu "articulos.css", el contingut del qual és:

```
nombre{color:blue;font-size:20px}  
precio{color:green;font-size:28px}
```

- **Escriu un document XML associat a l'arxiu "articulos.css" i que represente la següent informació:**

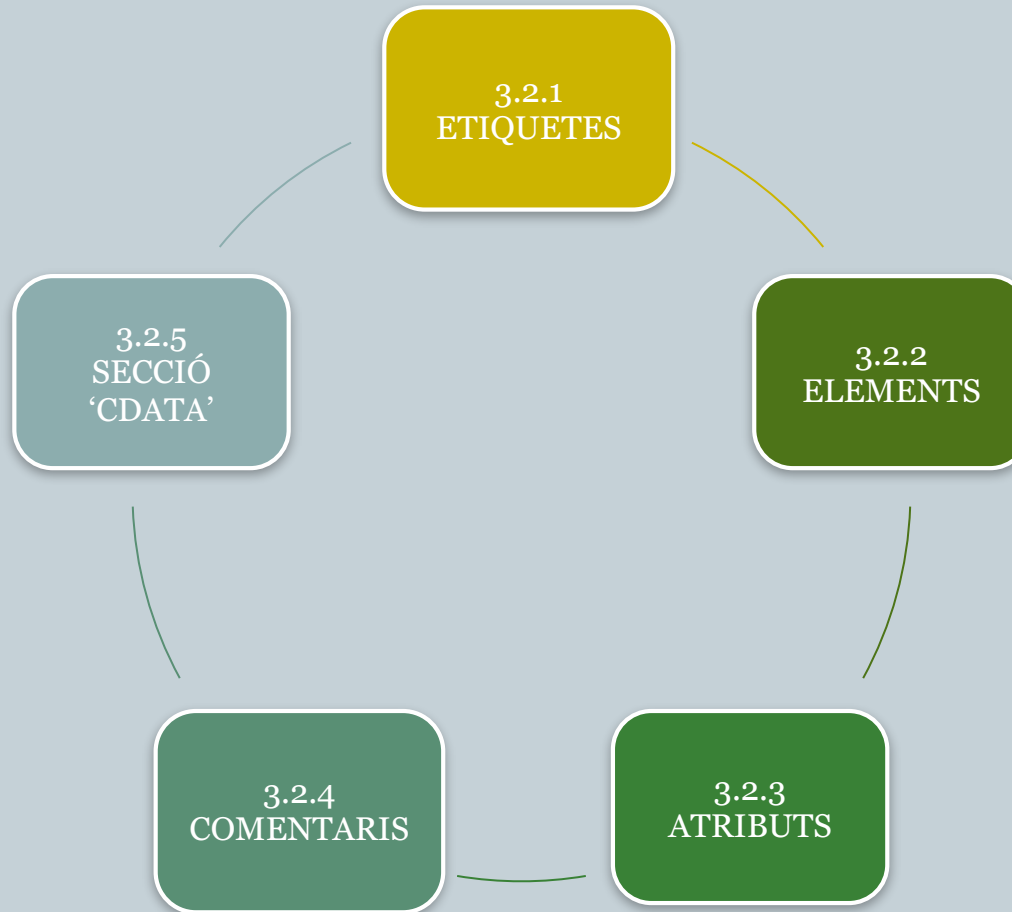
ARTÍCULOS	
Nombre	Precio
Sofá	642
Mesa	305
Lámpara	89

**El resultat al executar en el navegador serà el següent:**





## 3.2 Cos



## 3.2.1 Etiquetes



L'usuari pot definir les seues pròpies marques o etiquetes → guardar la informació que necessita i així separar el contingut de l'estructura → pilar fonamental en XML

### Tipus d'etiquetes:

- **Obertura** <casa>
- **Tancament** </casa>
- **Buides** (manquen de contingut) <casa/>

### A tindre en compte:

- Les etiquetes no poden contindre espais en blanc, cometes simples o dobles, punt i coma, signe de percentatge, etc.
- Són sensibles a les majúscules i minúscules (case sensitive).

## 3.2.2 Elements



Els elements estan compostos per etiquetes d'obertura i de tancament, escrites amb el mateix nom i tot el contingut que hi haja entre ambdues.

Poden existir elements buits (no és habitual) →

```
<casado></casado>
```

Un element buit pot ser un que no tinga etiqueta de tancament amb o sense atributs →

```
<teléfono n="666777888" />
```

### **A tindre en compte:**

- Els elements han d'estar niats correctament els uns dins dels altres, tancant-se en ordre invers al que s'obrin.
- Els elements no buits tindran etiquetes d'obertura i tancament.
- Els noms dels elements i atributs han de seguir una nomenclatura específica. Han de començar per lletres o els caràcters guió baix (\_) o dos punts (:), seguits de guions (mitjà o baix), punts, números o lletres. Tampoc poden començar per la paraula XML. El guió mitjà (-) i el punt (.) estan desaconsellats.

## 3.2.2 Elements



### EXERCICI PROPOSAT 3– Indica els elements que hi ha en el següent document XML

```
<feed xmlns="http://www.w3.org/2005/Atom">  
  <title>Árboles subtropicales</title>  
  <subtitle>cultivo del aguacate</subtitle>  
</feed>
```

### EXERCICI PROPOSAT 4– Indica els elements que hi ha en el següent document XML i dibuixa l'arbre amb la jerarquia que segueixen aquests elements.

```
<persona>  
  <nombre>Elsa</nombre>  
  <mujer/>  
  <fecha-de-nacimiento>  
    <día>18</día>  
    <mes>6</mes>  
    <año>1996</año>  
  </fecha-de-nacimiento>  
  <ciudad>Pamplona</ciudad>  
</persona>
```

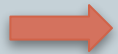
## 3.2.2 Elements



### Elements amb contingut mixt

#### EXAMPLE

```
<persona>  
  <nom>Elsa</nom> viu en <ciutat>Pamplona</ciutat>.  
</persona>
```



En aquest exemple, l'element "persona" conté els elements "nom" i "ciutat", a més dels textos " viu en" i ".".

## 3.2.3 Atributs



- S'usen per a assignar propietats associades als elements. Se situen dins d'una etiqueta d'obertura.
- S'especifiquen de la següent manera → NomAtribut="valor"
- Es poden afegir diversos atributs separats per un espai en blanc
- Els valors van entre cometes dobles o simples
- No pot haver-hi dos atributs amb el mateix nom
- Es poden utilitzar en XML però no és recomanable abusar d'ells, ja que el llenguatge XML ha de ser comprensible per a qualsevol usuari.

```
<nombre dni="77777777J"> AlumnoLM </nombre>
```

## 3.2.3 Atributs



**EXERCICI PROPOSAT 5– Crea un element “alumno” que continga dos atributs.**

**EXERCICI PROPOSAT 6 – Donat el següent fitxer XML d'una agenda personal, transforma els elements “ciudad” e “email” en atributs.**

```
<?xml version="1.0" encoding="UTF-8"?>
<agenda>
  <contacto nombre="alumno">
    <ciudad>Málaga</ciudad>
    <teléfono n="666777888"/>
    <email>a@librolm.com</email>
  </contacto>
</agenda>
```

## 3.2.4 Comentaris



Comentaris



```
<?xml version="1.0" encoding="utf-8" ?>
<agenda>
  <!--vamos a crear un contacto-->
  <contacto nombre="alumno LM">
    <ciudad>Málaga</ciudad>
  </contacto>
</agenda>
```



## 3.2.5 Secció CDATA



**Secció CDATA** → permet afegir contingut sense ser processat o analitzat. Han d'aparèixer dins de l'element arrel d'un document XML



```
<?xml version="1.0" ?>
<asignaturas>
  <foro>
    <![CDATA[el <b>examen</b>,será el día]]>
  </foro>
</asignaturas>
```

# 4. Elaboració d'un document XML ben format



**Per a poder elaborar un document XML ben format, és necessari seguir una sèrie de regles , entre les que destaquen:**

- El W3C recomana començar per la instrucció on s'indique la versió (instrucció de processament).
- Tot document ha de tindre una estructura jeràrquica , en la qual existisca un únic element arrel, que siga el primer a obrir-se i l'últim a tancar-se.
- Les etiquetes han d'estar niades correctament unes dins de les altres, tancant-se en ordre invers al que s'han obert. Els elements no buits, hauran de tindre tant etiqueta d'obertura com de tancament.
- Dins dels atributs, els valors han d'estar entre cometes dobles o simples.
- No poden haver dos atributs iguals en un mateix element.
- A l'hora d'escriure les etiquetes, elements o atributs, s'ha de tindre en compte que són 'case sensitive'.
- Els noms de les etiquetes, elements o atributs han de complir una sèrie de normes com les següents:  
No començar per un número, no començar amb la paraula XML, no usar caràcters especials reservats (>, <, @, &, ", etc.), començar sempre per una lletra i continuar per un guió, punt, número o altres lletres.
- Els comentaris no poden anar dins de les etiquetes.

# 4. Elaboració d'un document XML ben format

## EXEMPLE D'UN DOCUMENT BEN FORMAT

```
<?xml version="1.0" encoding="utf-8" ?>
<agenda>
  <!--vamos a crear un contacto-->
  <contacto>
    <nombre> AlumnoLM </nombre>
    <ciudad cp="29000">Málaga</ciudad>
    <teléfono n="666777888"/>
    <email>666777888@prueba.com</email>
    <casado v="no" />
  </contacto>
</agenda>
```

Sobre el mateix document, es realitzaran algunes modificacions per a veure el que mostra el navegador:

```
<?xml version="1.0" encoding="utf-8" ?>
<agenda>
  <!--vamos a crear un contacto-->
  <contacto>
    <nombre> AlumnoLM </Nombre>
    <ciudad cp="29000">Málaga</ciudad>
    <teléfono n="666777888"/>
    <9email>666777888@prueba.com</9email>
    <casado v="no" />
  </contacto>
</agenda>
```

# 4. Elaboració d'un document XML ben format



## RESULTATS:

```
<?xml version="1.0" encoding="utf-8" ?>
<agenda>
  <!--vamos a crear un contacto-->
  <contacto>
    <nombre> AlumnoLM </Nombre>
    <ciudad cp="29000">Málaga</ciudad>
    <teléfono n="666777888"/>
    <9email>666777888@prueba.com</9email>
    <casado v="no" />
  </contacto>
</agenda>
```

Error de lectura XML: etiqueta sin pareja. Se esperaba: </nombre>.  
Ubicación: file:///C:/Users/Sergio/Desktop/XMLFile5.xml  
Número de línea 5, columna 25:

```
<nombre> AlumnoLM </Nombre>
-----^
```

Error de lectura XML: mal formado  
Ubicación: file:///C:/Users/Sergio/Desktop/XMLFile5.xml  
Número de línea 8, columna 6:

```
<9email>666777888@prueba.com</9email>
-----^
```

# 4. Elaboració d'un document XML ben format



**EXERCICI PROPOSAT 7– Comprova si el següent document XML està ben format:**

```
<?xml version="1.0" encoding="utf-8"?>
<padre>
  <hijo>
    <nombre>alumno 1</nombre>
    <apellidos>ap1 ap2</apellidos>
  </hijo>
  <hijo>
    <nombres>alumno 2</nombre>
    <apellidos>ap11 ap21</apellidos>
  </hijo>
</padre>
```

**EXERCICI PROPOSAT 8– Donat el següent fitxer XML, introdueix un altre llibre i especifica l'element arrel de l'exemple. Una vegada afegit, comprova si està ben format.**

```
<?xml version="1.0" encoding="utf-8"?>
<biblioteca>
  <libro>
    <título>lenguajes de marcas</título>
    <autor>Autor 1</autor>
  </libro>
</biblioteca>
```