

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



**LLenguatges de marques i sistemes de  
gestió de la informació  
(1ºDAM)**

*Professora: Isabel Soriano*

# CONTINGUTS



## 1. INTRODUCCIÓ

## 2. GENERADORS D'ESQUEMES

## 3. ESQUEMES DTD (Document Type Definition)

3.1 ELEMENTS

3.2 ATRIBUTS

3.3 ENTITATS

3.4 ASSOCIACIÓ D'UN DTD A UN XML.

## 4. ESQUEMES XSD (XML Schema Definition)

4.1 ELEMENTS SIMPLES

4.2 ELEMENTS COMPLEXOS

4.3 ATRIBUTS

4.4 RESTRICCIONS

4.5 DOCUMENTACIÓ

4.6 ASSOCIACIÓ D'UN XSD A UN XML.

## 5. VALIDACIÓ D'UN DOCUMENT XML.

# 1. Introducció



Si volem especificar una sèrie de restriccions en un document XML, les opcions més utilitzades són els esquemes. Dos dels esquemes més utilitzats són:

- Mitjançant **DTD** (*Document Type Definition*) → senzill però poc flexible en les definicions dels elements.
- Mitjançant **XSD** (*XML Schema Definition*) → permet definir restriccions i elements més complexos.

## 2. Generadors d'esquemes



Existeixen generadors d'esquemes com Oxygen XML Editor, XMLSpy Visual Studio, XML Copy Editor, etc. (d'escriptori) i també en línia com els que es mostren a continuació.



<https://xml.mherman.org/>

**XML Schema Generator**

The XML Schema Generator creates a basic, easily adapted XML schema from an XML file.

Paste the contents of your XML file below: \*

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

Output Format: \*

**RNG**  **RNC**  **DTD**  **XSD**

**Generate**

This XML Schema Generator accepts one XML document and infers a schema.

It produces as output a schema written in any of the following formats:

- **RNG**: RELAX NG (XML syntax)
- **RNC**: RELAX NG (compact syntax)
- **DTD**: XML 1.0 DTD
- **XSD**: W3C XML Schema

**XML Schema Generator**

Generated Schema (.dtd):

```
<?xml encoding="UTF-8"?>
<!ELEMENT persona (nombre,mujer,fecha-de-nacimiento,ciudad)>
<!ATTLIST persona
  xmlns CDATA #FIXED ">

<!ELEMENT nombre (#PCDATA)>
<!ATTLIST nombre
  xmlns CDATA #FIXED ">

<!ELEMENT mujer EMPTY>
<!ATTLIST mujer
  xmlns CDATA #FIXED ">

<!ELEMENT fecha-de-nacimiento (dia,mes,año)>
<!ATTLIST fecha-de-nacimiento
  xmlns CDATA #FIXED ">

<!ELEMENT ciudad (#PCDATA)>
<!ATTLIST ciudad
  xmlns CDATA #FIXED ">

<!ELEMENT dia (#PCDATA)>
<!ATTLIST dia
  xmlns CDATA #FIXED ">

<!ELEMENT mes (#PCDATA)>
<!ATTLIST mes
  xmlns CDATA #FIXED ">
```

**Generate another XML Schema**

## 2. Generadors d'esquemes



<https://www.liquid-technologies.com/online-xml-to-xsd-converter>



Free Online XML to XSD Converter

Uses the sample XML document to infer an XML schema (XSD).

Document Valid

Download Free Liquid Studio Community Edition Now!

Sample XML Document

```
1 <!-- Add XML Data -->
2 <persona>
3   <nombre>Elsa</nombre>
4   <mujer>
5     <fecha-de-nacimiento>
6       <dia>18</dia>
7       <mes>6</mes>
8       <año>1996</año>
9     </fecha-de-nacimiento>
10    <ciudad>Pamplona</ciudad>
11  </persona>
12
```

Options

Generate Schema

Inferred XML Schema (XSD)

```
<?xml version="1.0" encoding="utf-8"?>
<!-- Created with Liquid Technologies Online Tools 1.0 (https://www.liquid-technologies.com) -->
<xss:schema attributeFormDefault="unqualified" elementFormDefault="qualified" xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="persona">
    <xss:complexType>
      <xss:sequence>
        <xss:element name="nombre" type="xs:string" />
        <xss:element name="mujer" />
        <xss:element name="fecha-de-nacimiento">
          <xss:complexType>
            <xss:sequence>
              <xss:element name="dia" type="xs:unsignedByte" />
              <xss:element name="mes" type="xs:unsignedByte" />
              <xss:element name="año" type="xs:unsignedShort" />
            </xss:sequence>
          </xss:complexType>
        </xss:element>
        <xss:element name="ciudad" type="xs:string" />
      </xss:sequence>
    </xss:complexType>
  </xss:element>
</xss:schema>
```

### 3. Esquemes DTD (*Document Type Definition*)



Un esquema DTD és un document de text pla on s'especifiquen o defineixen una sèrie de normes que s'usen per a saber com ha de crear-se un fitxer XML per a poder validar-lo. El DTD defineix i indica tant els elements que intervenen com les regles a complir en un document XML.

La creació d'un DTD no és obligatòria però sí recomanable →principalment en casos en els quals la informació del XML vaja a ser compartida.

L'extensió ha de ser .dtd

## 3.1 Elements



Els elements d'un DTD són la base de l'estructura. S'indiquen de la següent manera:

- (1) Primer s'especifica la dependència jeràrquica entre el node pare i els fills.

```
<!ELEMENT nombreElemento (elementos_hijo1, elemento_hijo2, ...)>
```

- (2) Més tard, s'indica per a cada element el tipus.

```
<!ELEMENT nombreElemento Tipo>
```



### Els tipus poden ser:

- **(#PCDATA)** → *Parsed Character Data* → El contingut és text i analitzat per un parser.
- **(#CDATA)** → *Character Data* → El contingut no és analitzat per un parser i no poden detectar-se possibles entitats.
- **ANY** → Els elements poden contindre qualsevol valor.
- **EMPTY** → No té contingut.

Un **parser** és un analitzador o processador XML que comprova si el document està ben format o és vàlid analitzant la seua estructura.

# 3.1 Elements



## Ocurredència d'aparició dels elements:

Caràcter	Descripció
?	L'element pot aparéixer zero o una vegada (opcional)
*	L'element pot aparéixer zero, una o diverses vegades (opcional)
+	L'element ha d'aparéixer una o més vegades (obligatori)
	Similar al OR lògic, indica que pot afegir-se un dels dos elements que s'especifiquen, però no tots dos (obligatori).

```
<!ELEMENT NombreE (hijo1, (hijo2 | hijo3)) >
```

# 3.1 Elements



## EXAMPLE

DTD

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE BDcomponentes [
<!ELEMENT BDcomponentes (componente)+>
<!ELEMENT componente (nombre,(precio|precioiva),tipo+,medida)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT precioiva (#PCDATA)>
<!ELEMENT tipo (#PCDATA)>
<!ELEMENT medida (#PCDATA)>
]>
<BDcomponentes>
  <componente>
    <nombre>resistencia</nombre>
    <precio>0.25</precio>
    <tipo>pasivo</tipo>
    <tipo>bobinadas</tipo>
    <medida>ohmios</medida>
  </componente>
  <componente>
    <nombre>condensador</nombre>
    <precioiva>0.75</precioiva>
    <tipo>pasivo</tipo>
    <medida>Faradios</medida>
  </componente>
</BDcomponentes>
```

XML

# 3.1 Elements



## EXAMPLE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE BDcomponentes [
<!ELEMENT BDcomponentes (componente)+>
<!ELEMENT componente (nombre,(precio|precioiva),tip+,medida)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
<!ELEMENT precioiva (#PCDATA)>
<!ELEMENT tipo (#PCDATA)>
<!ELEMENT medida (#PCDATA)>
]>
<BDcomponentes>
<componente>
  <nombre>resistencia</nombre>
  <precio>0.25</precio>
  <tipo>pasivo</tipo>
  <tipo>bobinadas</tipo>
  <medida>ohmios</medida>
</componente>
<componente>
  <nombre>condensador</nombre>
  <precioiva>0.75</precioiva>
  <medida>Faradios</medida>
</componente>
</BDcomponentes>
```

Si ho definim amb (+) ha d'aparéixer obligatòriament el <tipo>. Perquè no siga un error i puga no aparéixer cap vegada aqueix element, es posaria (\*) en comptes de (+)

# 3.1 Elements



**EXERCICI PROPOSAT 1– A partir del DTD i de la informació de la taula, crea el fitxer XML corresponent.**

Campos de un *email*

Receptor	Asunto	Cuerpo
r1@librolm.com	Cumpleaños	Desearte un feliz...
r2@librolm.com	Día libre	Estimado señor X,...

Un fichero DTD para un XML podría ser:

```
<!ELEMENT bdcorreo (email)+>
<!ELEMENT email (receptor,asunto,cuerpo)>
<!ELEMENT receptor (#PCDATA)>
<!ELEMENT asunto (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

## 3.2 Atributs



- S'usen per a enriquir o afegir informació a un element.
- Si un element té més d'un atribut es defineixen un darrere d'un altre separats per un espai en blanc, tabulació salte de línia.
- Es declaren de la següent manera:

```
<!ATTLIST Elemento Atributo Tipo Valor>
```

Valor	Descripció
#FIXED	Ha d'especificar-se un valor que és fix i no podrà canviar-se.
#IMPLIED	L'atribut es opcional.
#DEFAULT	Ha d'especificar-se un valor per defecte que sí que podrà canviar-se.
#REQUIRED	L'atribut es obligatori.

Tipus	Descripció
CDATA	El valor és alfanumèric (cadena de text).
Enumeracions	Permet especificar diversos valors per a un atribut.

\*\* Hi ha més tipus d'atributs, però els que s'indiquen en la taula són els més utilitzats.

### EXEMPLE Enumeració

```
<!ATTLIST numero octal(0|1|2|3|4|5|6|7) CDATA #REQUIRED>
```

## 3.2 Atributs



### EXEMPLE

DTD

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE bdusuarios [
  <!ELEMENT bdusuarios (usuario)+>
  <!ELEMENT usuario (nombre,email,fechaNac)>
  <!ATTLIST usuario
    clave CDATA #REQUIRED
    dni CDATA #REQUIRED>

  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT fechaNac EMPTY>
]>
<bdusuarios>
  <usuario clave="**" dni="*****">
    <nombre>Ataulfo G. Pascual</nombre>
    <email>user1@librolm.com</email>
    <fechaNac></fechaNac>
  </usuario>
  <usuario clave="****" dni="*****">
    <nombre>Francisco J. García</nombre>
    <email>user2@librolm.com</email>
    <fechaNac></fechaNac>
  </usuario>
</bdusuarios>
```

XML

2 atributs obligatoris de tipus text.

## 3.3 Entitats

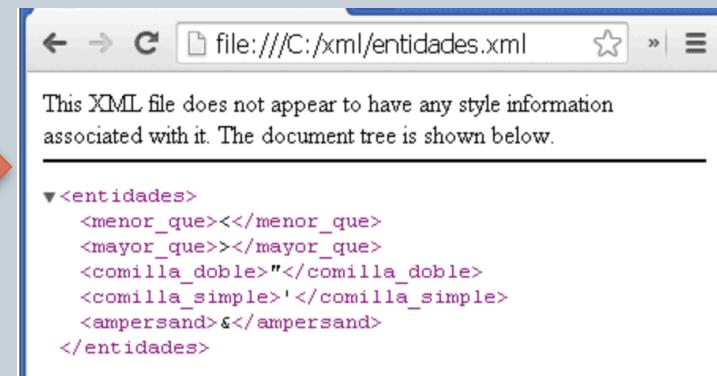
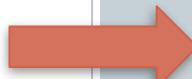


Les entitats es consideren constants i poden usar-se per a abreujar text o utilitzar alguns caràcters especials com els que es mostren en la següent taula:

Caràcter	Entitat	Referència a entitat
< (menor que)	lt (less than)	&lt;
> (major que)	gt (greater than)	&gt;
“ (cometa doble)	quot (quotation mark)	&quot;
‘ (cometa simple)	apos (apostrophe)	&apos;
& (ampersand)	amp (ampersand)	&amp;

### EXEMPLE

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



The XML file does not appear to have any style information associated with it. The document tree is shown below.

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

# 3.3 Entitats



- Es poden classificar en INTERNES i EXTERNES (especificant URL que conté l'entitat).
- Per a referenciar a un entitat dins del XML → **&entidad;**
- Per a crear una entitat dins d'un DTD → `<!ENTITY nombre valor>`

## EXEMPLE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE bdusuarios[
  !ELEMENT bdusuarios (usuario)+>
  !ELEMENT usuario (email,nombre,suceso)>
  !ELEMENT email (#PCDATA)>
  !ELEMENT nombre (#PCDATA)>
  !ELEMENT suceso (#PCDATA)>
  !ENTITY web "http://www.librolm.com">
  !ENTITY cliente "Se informa que el usuario ha accedido al">
]>
<bdusuarios>
  <usuario>
    <email>user1@librolm.com</email>
    <nombre>Ataulfo G. Pascual</nombre>
    <suceso> &cliente; registro en la página &web; </suceso>
  </usuario>
  <usuario>
    <email>user2@librolm.com</email>
    <nombre>Francisco J. García</nombre>
    <suceso> &cliente; contenido de &web; para consultar el libro X</
      suceso>
  </usuario>
</bdusuarios>
```



En el navegador:

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<!DOCTYPE bdusuarios (View Source for full doctype...)>
- <bdusuarios>
  - <usuario>
    <email>user1@librolm.com</email>
    <nombre>Ataulfo G. Pascual</nombre>
    <suceso>Se informa que el usuario ha accedido al registro en la
      página http://www.librolm.com</suceso>
  </usuario>
  - <usuario>
    <email>user2@librolm.com</email>
    <nombre>Francisco J. García</nombre>
    <suceso>Se informa que el usuario ha accedido al contenido de
      http://www.librolm.com para consultar el libro X</suceso>
  </usuario>
</bdusuarios>
```

## 3.4 Associació d'un DTD a un XML



Es pot vincular un fitxer DTD a un XML de manera **INTERNA** (en el mateix document XML, al principi) o **EXTERNA** (recomanable si una DTD s'usa per a diferents XML i fora necessari fer modificacions).

**De manera INTERNA** →

la segona línia del XML serà: **<!DOCTYPE Element\_arrel [Elements]>**

**De manera EXTERNA** →

la segona línia del XML serà: **<!DOCTYPE Element\_arrel definició “fitxer.dtd”>**

*\*\* La definició pot ser privada (SYSTEM) que és la més habitual o pública (PUBLIC) definida per organismes.*

# 3.4 Associació d'un DTD a un XML



## EXEMPLE

### DTD INTERN

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE bdcorreo [
  <!ELEMENT bdcorreo (email)+>
  <!ELEMENT email (receptor,asunto,cuerpo)>
  <!ELEMENT receptor (#PCDATA)>
  <!ELEMENT asunto (#PCDATA)>
  <!ELEMENT cuerpo (#PCDATA)>
]>
<bdcorreo>
  <email>
    <receptor>r1@librolm.com</receptor>
    <asunto>Cumpleaños</asunto>
    <cuerpo>Desearte un feliz..</cuerpo>
  </email>
  <email>
    <receptor>r2@librolm.com</receptor>
    <asunto>Día libre</asunto>
    <cuerpo>Estimado señor X,...</cuerpo>
  </email>
</bdcorreo>
```

### DTD EXTERN

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE bdcorreo SYSTEM "midtd.dtd">
<bdcorreo>
  <email>
    <receptor>r1@librolm.com</receptor>
    <asunto>Cumpleaños</asunto>
    <cuerpo>Desearte un feliz..</cuerpo>
  </email>
  <email>
    <receptor>r2@librolm.com</receptor>
    <asunto>Día libre</asunto>
    <cuerpo>Estimado señor X,...</cuerpo>
  </email>
</bdcorreo>
```

```
<!ELEMENT bdcorreo (email)+>
<!ELEMENT email (receptor,asunto,cuerpo)>
<!ELEMENT receptor (#PCDATA)>
<!ELEMENT asunto (#PCDATA)>
<!ELEMENT cuerpo (#PCDATA)>
```

## 3.4 Associació d'un DTD a un XML



**EXERCICI PROPOSAT 2– Donat el DTD per a crear diferents espècies d'arbres, obtingues el XML associat:**

```
<!ELEMENT árbol (especie)+>
<!ELEMENT especie (nombre, variedad, (origen)+, color_fruto,
    (maduración)+, (riego)?, (precio)?)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT variedad (#PCDATA)>
<!ELEMENT origen (#PCDATA)>
<!ELEMENT color_fruto (#PCDATA)>
<!ELEMENT maduración (#PCDATA)>
<!ELEMENT riego (#PCDATA)>
<!ELEMENT precio (#PCDATA)>
```

# 4. Esquemes XSD (*XML Schema Definition*)



```
<?xml version= "1.0" encoding= "utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    ...
    Elementos
    ...
</xs:schema>
```

- Fitxers amb extensió .xsd
- **Element arrel** → <**xs:schema**> </**xs:schema**> (xs és el prefix utilitzat, també se sol usar xsd)
- En crear un XML schema fem ús dels elements i atributs especificats en l'estàndard de XML Schema. Perquè això siga possible hem d'incloure en l'element arrel de l'esquema (l'element “schema”) **una referència a l'espai de noms “http://www.w3.org/2001/XMLSchema”**. Això es fa incloent l'atribut “xmlns” en el element “schema”.  
→ xmlns: **xs**=“http://www.w3.org/2001/XMLSchema”
- **Declarar elements** → elementFormDefault=“qualified/unqualified”, indica si ha d'afegir-se l'espai de noms davant o no.
- **Declarar atributs** → attributeFormDefault=“qualified/unqualified”
- **Versió del document d'esquema** → versión

# 4.1 Elements simples



## 2 tipus d'elements: simples (sense fills ni atributs) i complexos

### ELEMENTS SIMPLES

- Es defineixen amb l'etiqueta <xs:element>, contenint sol valors d'un determinat tipus.

```
<xs:element Nombre Tipo Valor/>
```

### EXEMPLE

```
<xs:element name="color" type="xs:string" default="azul"/>
```

**Valor** → default (valor per defecte que pot ser modificat) o fixed (no podrà canviar-se).

### TIPUS DE DADES PREDEFINIDES EN XSD

Tipo	Descripción
decimal	Número decimal.
boolean	Un único valor lógico (1, 0).
dateTime	Fecha y hora en formato (AAAA-MM-DD T HH:MM:SS).
string	Texto o cadena de caracteres.
date	Fecha: año-mes-día (aaaa-mm-dd).
time	Hora: hh:mm:ss.
integer	Número entero positivo o negativo.
positiveInteger	Número entero positivo.
long	Entero de 64 bits.
short	Entero de 16 bits.

## 4.1 Elements simples



### **Creació de tipus de dades simples personalitzades**

- Poden ser assignats a elements o atributs.
- Es poden reutilitzar en l'esquema XSD.
- L'atribut 'name' serveix per a assignar-li'l a l'element i indicar que és un tipus simple nou.

```
<xs:simpleType name="Tcat1">  
  ...  
</xs:simpleType>
```

## 4.1 Elements simples



### Indicadors d'ocurrència per a un element - Cardinalitat

Atribut	Descripció
<b>maxOccurs</b>	S'especifica un valor amb el nombre màxim de vegades que pot aparéixer l'element, el valor del qual per defecte és 1. Per a especificar un nombre de vegades il·limitat, el valor és "unbounded".
<b>minOccurs</b>	S'especifica un valor amb el nombre mínim de vegades que pot aparéixer l'element, el valor del qual per defecte és 1. Per a especificar un nombre de vegades il·limitat, el valor és

# 4.1 Elements simples



**EXERCICI PROPOSAT 3– Per als següents elements del document XML, escriu com serien les seues definicions d'elements simples corresponents.**

## XML

```
<ciudad>Roma</ciudad>

<fecha-de-nacimiento>1996-12-18</fecha-de-nacimiento>

<hora>18:29:45</hora>

<nota>7.5</nota>

<apto>true</apto>
```

**EXERCICI PROPOSAT 4– Definir un element anomenat *puertaCerrada* de tipus lògic, que per defecte tinga el valor "fals", i un altre element anomenat *ventanaAbierta* també de tipus lògic, que tinga assignat el valor fix "vertader".**

## 4.2 Elements complexos



**Els elements complexos contenen atributs i altres elements fills. La seua estructura bàsica és la següent:**

```
<xs:element name="nombreElementoCompuesto">
  <xs:complexType>
    <Indicador_Orden>
      Elementos / atributos
    </ Indicador_Orden>
  </xs:complexType>
</xs:element>
```

## 4.2 Elements complexos



Quan es defineix un element simple s'ha de donar informació sobre el seu tipus de dades. Fins ara, hem vist que existeixen tipus predefinits com 'integer', 'boolean' o 'string'. No obstant això, també podem, com usuaris, definir els nostres propis tipus de dades. Per tal d'aconseguir-ho, s'utilitzen els elements complexos. En aquests tipus d'elements, s'ha de donar un nom (atribut 'name'), de manera que puguen ser referenciats en altres punts de l'esquema.

### EXEMPLE

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="usuario" type="Tuser"/> Tipus de dades creat amb l'element complex
  <xs:complexType name="Tuser" >
    <xs:sequence>
      <xs:element name="nombre" type="xs:string" />
      <xs:element name="email" type="xs:string" />
      <xs:element name="fechaNac" type="xs:date" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

*Elements simples amb tipus de dades predefinides.*

## 4.2 Elements complexos



**Quan són elements complexos, s'ha d'especificar l'ordre d'aparició dels diferents elements en el document XML (en DTD no era possible):**

- <xs:sequence>
- <xs:choice>
- <xs:all>

```
<xs:element name="nombreElementoCompuesto">
  <xs:complexType>
    <Indicador_Orden>
      Elementos / atributos
    </ Indicador_Orden>
  </xs:complexType>
</xs:element>
```



## 4.2 Elements complexos



**<xs:sequence>** → s'especifiquen els fills de l'element principal, seguint la seqüència o ordre d'aparició indicada.

### EXEMPLE D'ESQUEMA XSD.

```
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="usuario">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="email" type="xs:string"/>
        <xs:element name="fechaNac" type="xs:date"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## 4.2 Elements complexos



**Seria VÀLID el següent XML (segons el XSD anterior)?**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <nombre>Francisco J. García</nombre>
  <fechaNac>2018-10-25</fechaNac>
  <email>user2@librolm.com</email>
</usuario>
```



**NO, l'ordre no és correcte.**

**SOLUCIÓ**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <nombre>Francisco J. García</nombre>
  <email>user2@librolm.com</email>
  <fechaNac>2018-10-25</fechaNac>
</usuario>
```

## 4.2 Elements complexos



**<xs:choice>** —> permet especificar un dels elements fills que apareixen, però només un d'ells.

### EXEMPLE D'ESQUEMA XSD.

```
<xs:element name="usuario">
  <xs:complexType> [...] ...
    <xs:choice>
      <xs:element name="nombreU" type="xs:string"/>
      <xs:element name="email" type="xs:string"/>
    </xs:choice>
  </xs:complexType>
</xs:element>
</xs:schema>
```

## 4.2 Elements complexos



**Seria VÀLID el següent XML (segons el XSD anterior)?**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <nombreU>Sergio</nombreU>
  <email>user2@librolm.com</email>
</usuario>
```



**NO, només pot aparéixer  
un dels elements fills.**

**POSSIBLE SOLUCIÓ**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <email>user2@librolm.com</email>
</usuario>
```

## 4.2 Elements complexos



**<xs:all>** —> permet especificar la seqüència d'elements en qualsevol ordre, encara que sense poder afegir dins d'ell **<xs:choice>** o **<xs:sequence>**. Han d'aparéixer tots els elements.

### EXEMPLE D'ESQUEMA XSD.

```
<xs:schema elementFormDefault="qualified" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="usuario">
    <xs:complexType>
      <xs:all>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="email" type="xs:string"/>
        <xs:element name="fechaNac" type="xs:date"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## 4.2 Elements complexos



**Seria VÀLID el següent XML (segons el XSD anterior)?**

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <fechaNac>2018-10-25</fechaNac>
  <email>user2@librolm.com</email>
</usuario>
```



**NO, han d'aparéixer tots els elements.**



**POSSIBLES SOLUCIONS**

```
?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <nombre>Francisco J. García</nombre>
  <fechaNac>2018-10-25</fechaNac>
  <email>user2@librolm.com</email>
</usuario>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <fechaNac>2018-10-25</fechaNac>
  <nombre>Francisco J. García</nombre>
  <email>user2@librolm.com</email>
</usuario>
```

```
<?xml version="1.0" encoding="iso-8859-1"?>
<usuario>
  <fechaNac>2018-10-25</fechaNac>
  <email>user2@librolm.com</email>
  <nombre>Francisco J. García</nombre>
</usuario>
```

## 4.3 Atributs

### EXEMPLE

```
<xs:attribute NombreA TipoA USO ValorA/>
```

**ValorA** → default (valor per defecte que pot ser modificat) o fixed (no podrà canviarse)

```
<xs:attribute name="dni" type="xs:integer" use="required"/>
```

Indiquem use="required" per a indicar que és obligatori. Si no posem res, per defecte seria use="optional".

### EXERCICI PROPOSAT 5- Realitza l'esquema XSD del següent XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns: xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha numero="1">
    <nombre>Ana Sanz Tin</nombre>
    <edad>22</edad>
  </ficha>
  <ficha numero="2">
    <nombre>Iker Rubio Mol</nombre>
    <edad>23</edad>
  </ficha>
</fichas>
```

## 4.4 Restriccions



Restricció	Descripció
<b>length</b>	Conté un atribut value on s'especifica un valor de longitud fixa.
<b>minLength</b>	Conté un atribut value on s'especifica el valor de longitud mínima.
<b>maxLength</b>	Conté un atribut value on s'especifica el valor de longitud màxima.
<b>whiteSpace</b>	Conté un atribut value on s'especifica un valor que pot ser “preserve”, “replace”, “collapse”. Tracta l'ús dels espais en blanc, retorns de carro, etc., que puguen aparéixer.
<b>maxExclusive</b>	Conté un atribut value on s'especifica el valor, que és menor a l'especificat.
<b>minExclusive</b>	Conté un atribut value on s'especifica el valor, que és major a l'especificat.
<b>minInclusive</b>	Conté un atribut value on s'especifica el valor, que és major o igual a l'especificat.
<b>maxInclusive</b>	Conté un atribut value on s'especifica el valor, que és menor o igual a l'especificat.
<b>totalDigits</b>	Conté un atribut value on s'especifica el valor, que és el nombre màxim de díigits d'un número tenint en compte els decimals.
<b>fractionDigits</b>	Conté un atribut value on s'especifica el valor, que és el nombre màxim de díigits de decimals d'un número.
<b>enumeration</b>	Conté un atribut value on s'especifica un dels valors admesos de la llista.
<b>pattern</b>	Conté un atribut value on s'especifica un rang de caràcters admesos o expressió regular.

## 4.4 Restriccions



### EXERCICI PROPOSAT 6- Crear el XSD associat al document XML per a emmagatzemar dades relatives a diversos productes.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<bdproductos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:no-
    NamespaceSchemaLocation="zax.xsd">
    <producto>
        <nombre>teclado</nombre>
        <codigo>C001</codigo>
        <categoria>A</categoria>
        <iva>21</iva>
        <precio>15</precio>
        <descuento>10</descuento>
    </producto>
    <producto>
        <nombre>monitor</nombre>
        <codigo>C002</codigo>
        <categoria>B</categoria>
        <iva>21</iva>
        <precio>147.2</precio>
        <descuento>9</descuento>
    </producto>
</bdproductos>
```

# 4.4 Restriccions

**EXERCICI PROPOSAT 7- Modifica el XSD per a afegir els següents tipus de dades personalitzades i les seues restriccions (s'han d'incloure al final abans de tancar l'esquema) .**

## RESTRICCIÓ NUMÈRICA

### DESCUENTO

```
<xs:simpleType name="Tdescuento">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="10"/>
  </xs:restriction>
</xs:simpleType>
```

### CÓDIGO

```
<xs:simpleType name="Tlongitudentre">
  <xs:restriction base="xs:string">
    <xs:minLength value="1"/>
    <xs:maxLength value="4"/>
  </xs:restriction>
</xs:simpleType>
```

### PRECIO

```
<xs:simpleType name="Tdecimal">
  <xs:restriction base="xs:decimal">
    <xs:totalDigits value="4"/>
    <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>
```

## RESTRICCIÓ LONGITUD DEL CONTINGUT

### NOMBRE

```
<xs:simpleType name="Tlongitud">
  <xs:restriction base="xs:string">
    <xs:length value="4"/>
  </xs:restriction>
</xs:simpleType>
```

## RESTRICCIÓ D'ENUMERACIÓ

### CATEGORÍA

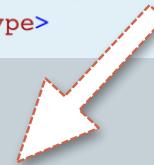
```
<xs:simpleType name="Tcat">
  <xs:restriction base="xs:string">
    <xs:enumeration value="A"/>
    <xs:enumeration value="B"/>
    <xs:enumeration value="C"/>
    <xs:enumeration value="D"/>
    <xs:enumeration value="E"/>
  </xs:restriction>
</xs:simpleType>
```

## 4.4 Restriccions



### RESTRICCIÓ MITJANÇANT EXPRESSIONS REGULARS O PATRONS

```
<xs:simpleType name="Tcat1">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-E]" />
  </xs:restriction>
</xs:simpleType>
```



[0-9]: número del 0 al 9.

[a-z]: letra minúscula perteneciente a ese intervalo.

[A-Z]: letra mayúscula perteneciente a ese intervalo.

[aeiou]: un carácter de los que aparecen entre los corchetes.

[^aeiou]: un carácter que no aparezca entre corchetes.

{X}: las llaves indican que tiene que repetirse X veces el contenido que haya delante.

### EXERCICI PROPOSAT 8- Com podríem elaborar un patró per al DNI?

## 4.5 Documentació



Si necessita dotar-se d'informació als esquemes perquè puga ser consultada per altres usuaris, s'usen les següents instruccions:

- **xs:annotation** —> especifica els comentaris dins d'un esquema que actuen com a documentació d'aquest. Té un atribut opcional (id) i dos elements fills (appinfo i documentation).
- **xs: appinfo** —> s'usa per a especificar la informació de l'aplicació.
- **xs: documentation** —> afeg comentaris en un esquema. Ha d'anar dins de 'xs:annotation' i té dos atributs opcionals (source i xml\_lang)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:annotation>
    <xs:appinfo>Lenguaje de marcas</xs:appinfo>
    <xs:documentation xml:lang="es">
      comentarios sobre
    </xs:documentation>
  </xs:annotation>
  ...

```

## 4.6 Associació d'un XSD a un XML



L'associació d'un document XSD a un XML es realitza mitjançant un espai de noms amb una sèrie d'atributs:

- **xmlns** —> definir l'espai de noms. Els elements de l'esquema portaran el prefix que es definisca, en aquest cas serà xs.

D'altra banda, les referències als esquemes poden realitzar-se mitjançant els següents atributs:

- **noNamespaceSchemaLocation** —> s'empra quan no s'utilitza espai de noms en el document i el que es fa és usar un fitxer amb l'esquema (.xsd)
- **schemaLocation** —> s'empra quan s'utilitzen explícitament els noms dels Espais de noms en les etiquetes.

# 4.6 Associació d'un XSD a un XML

## EXEMPLE

### XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<bdalumnos xmlns:xs="http://www.w3.org/2001/XMLSchema-instance" xs:noNa-
mespaceSchemaLocation="xsdalumnos.xsd">
  <alumno>
    <nombre>José Antonio</nombre>
    <padre>José Luis</padre>
    <madre>Carmen</madre>
    <edad>18</edad>
    <ciudad>Málaga</ciudad>
  </alumno>
  <alumno>
    <nombre>Francisco Jesús</nombre>
    <padre>José</padre>
    <madre>María</madre>
    <edad>25</edad>
    <ciudad>15</ciudad>
  </alumno>
</bdalumnos>
```

### xsdalumnos.xsd

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="bdalumnos">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" name="alumno">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string" />
              <xs:element name="padre" type="xs:string" />
              <xs:element name="madre" type="xs:string" />
              <xs:element name="edad" type="xs:unsignedByte" />
              <xs:element name="ciudad" type="xs:string" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

# 5. Validació d'un document XML

Un document pot estar ben format (sintaxi) però no ser vàlid (no compleix l'especificat en l'esquema DTD o XSD). En canvi, un document vàlid (compleix tant la sintaxi com l'estructura) sempre està ben format.

## EXAMPLE en XML Copy Editor

