

# XML/DTD

ORENGA RAMON, ANDREU

1º DAW/DAM

## CONTENIDO

¿Qué es XML?	3
Principales características de XML	3
Xml no es	3
En XML	4
Análisis XML	4
Errores en XML	4
Analizadores XML	4
Editores XML	4
XML Reglas Básicas	5
Reglas básicas	5
Etiqueta raíz:	5
Las etiquetas se tienen que cerrar	5
Correctamente anidadas	5
Nombres correctos	5
Case sensitive	5
Valores entre comillas (" ")	5
XML Creación de Documentos	6
Declaración	6
Etiquetas	6
Atributos	6
Entidades	7
CDATA	7
Comentarios	8
Instrucciones de proceso	8
Como asociar un archivo css a un documento xml	8
Espacios de nombres (namespace)	8
Esquemas y vocabularios en xml	9
DTD	9
Sintaxis basica	9
DTD Interno	10
DTD externo	10
Estructura DTD	10
Declaracion de elementos	10
Tipos de contenidos	11
Modelo de contenidos	11

Quantificadores .....	11
Atributos.....	12
Declaracion de atributos en dtd .....	12
Tipos de atributos.....	13
Valores de atributos .....	13
Valores predeterminados de atributos .....	13
Entidades.....	13
Uso de las entidades.....	13
Ventajas de las entidades.....	14
Entidades internas.....	14
Entidades generales externas analizadas .....	14
Entidades no analizadas .....	14
Ejemplo DTD .....	15

# XML/DTD

## ¿QUÉ ES XML?

Es un lenguaje de marcas utilizado para almacenar y intercambiar datos estructurados de manera legible tanto por humanos como por máquinas.

## PRINCIPALES CARACTERÍSTICAS DE XML

- Es un lenguaje de marcas: Utiliza etiquetas para definir la estructura de los datos. Elementos que envuelven los datos y describen lo que representan. <persona> (envuelve los datos de una persona).
- Extensible: Que podemos definir nuestras propias etiquetas y estructuras de datos.
- Jerarquizado: Sigue una estructura jerárquica, las etiquetas pueden contener otras etiquetas, creando una estructura en forma de árbol
- Legible para los humanos: Esta en un formato legible para los humanos. Facilita la inspección y la edición manual.
- Interoperabilidad: es un estándar altamente aceptado, comprensible por muchas aplicaciones y plataformas diferentes. Facilita el intercambio de datos entre sistemas diversos.
- Usos variados: se hace servir en una amplia gama de aplicaciones, bases de datos, documentos html, configuraciones de software etc.
- Compatible con Unicode: puede almacenar textos en diversas codificaciones incluyendo Unicode.
- Es versátil y está estandarizado

## XML NO ES

- Un lenguaje de programación.
- No es un protocolo de red.
- No es una base de datos.

Xml no es simplemente un sistema de marcas para estructurar datos, sino que también incluye tecnologías relacionadas que facilitan la creación, el almacenamiento, la transmisión y el procesamiento de datos.

- XPath y XQuery: utilizado para seleccionar y consultar datos dentro de documentos xml.
- XPointer: identifica partes específicas de un documento xml.
- XLink: proporciona una estructura para añadir enlaces de hipertexto.
- DTD: define la estructura de validación, permite asegurar que los datos siguen una estructura específica
- XSL: se utiliza para transformar documentos xml en otros formatos, como HTML.

## EN XML

- Los datos se pueden representar en árboles.
- Todos los documentos tienen una etiqueta padre (raíz) que no es hijo de nadie.
- Las hojas del árbol son los datos contenidos en el documento.
- Están formados por árboles de elementos. Empieza por un elemento raíz, del cual van saliendo las ramas, todos los elementos pueden tener elementos hijos.
- Todos los elementos pueden contener contenido textual y atributos.

## ANÁLISIS XML

El objetivo principal de tener reglas es que pueden ser leídas y interpretadas por un ordenador, se utilizan analizadores:

- Comprueban que se cumplen las reglas del xml.
- Comprueban que el documento este “bien formado”.
- Comprueban que el documento es “valido”.

Hay 2 operaciones de comprobación:

1. Correcto: comprueba que el documento sigue las reglas de creación (bien formado).
2. Valido: que el documento cumpla la estructura específica definida en un esquema o DTD.

## ERRORES EN XML

Errores leves: violación de algunas normas que están marcadas como recomendación (el programa puede intentar continuar).

Errores graves: no se cumplan algunas de las reglas obligatorias: no seguir la sintaxis etc... El programa tendrá que detenerse inmediatamente.

## ANALIZADORES XML

- XmlLint
- Apache Xerces
- Expat
- Waindows.Data.XML

## EDITORES XML

- Se pueden crear con cualquier editor de texto.
- Servirá de ayuda hacer servir algún editor con soporte xml (autocompletado, validación, creación de esquemas etc).
- Hay editores específicos para trabajar en xml (XML Copy Editor)
- Muchas veces IDEs de programación tienen soporte ( IntelliJ, Eclipse, Webstorm etc.)
- Si tenemos que hacer ediciones complejas podemos hacer servir editores especializados (oXigen XML, Liquid XML Studio, Stylus 6 XML).

## XML REGLAS BÁSICAS

Texto, etiquetas y atributos

```
<nombre cargo = "coordinadora"> Isabel S. </nombre>
```

- Etiqueta: <nombre></nombre>
- Atributo: cargo=" coordinadora"
- Texto: Isabel S.

No se propone nada sobre cómo se utilizarán los datos, el usuario tiene el control.

## REGLAS BÁSICAS

### ETIQUETA RAÍZ:

- Es la etiqueta que contiene todas las demás .
- Lo único que puede estar fuera de la raíz son los comentarios y las instrucciones de proceso.

### LAS ETIQUETAS SE TIENEN QUE CERRAR

- Cada una de las etiquetas se tiene que cerrar ( </etiqueta> )
- Si tenemos en etiquetas sin datos el cierre será />

### CORRECTAMENTE ANIDADAS

<ul style="list-style-type: none"><li>• Si es comencen dos etiquetas aquestes s'han de tancar correctament</li></ul>	<ul style="list-style-type: none"><li>• No es poden tancar de qualsevol forma (al estar tancat en ordre invers és incorrecte)</li></ul>
<pre>&lt;inventari&gt;   &lt;article&gt;     Disquet   &lt;/article&gt; &lt;/inventari&gt;</pre>	<pre>&lt;inventari&gt;   &lt;article&gt;     Disquet   &lt;/inventari&gt; &lt;/article&gt;</pre>

### NOMBRES CORRECTOS

- Tienen que empezar por una letra del alfabeto, subrayado, guion o dos puntos.
- No pueden tener espacios.
- No pueden empezar por la palabra xml.

### CASE SENSITIVE

Esto indica que xml diferencia entre las letras mayúsculas de las minúsculas.

### VALORES ENTRE COMILLAS (" ")

- Los valores de los atributos tienen que estar entre comillas.
- Los atributos siempre tienen que tener algún valor
- No importa si son comillas simples o dobles.

## XML CREACIÓN DE DOCUMENTOS

### DECLARACIÓN

La declaración es una línea especial que se introduce al principio del documento XML para identificar su versión y codificación de caracteres.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

La declaración no es obligatoria pero si recomendable. En muchos casos solo contendrá la versión y el encoding.

- <?XML: indica que se trata de un documento XML.
- Versión: generalmente se utiliza la versión 1.0.
- Encoding: define el código de caracteres que hacemos servir, se hace servir Unicode por defecto (UTF-8 y UTF-16) pero se pueden utilizar otros:

- **xmlns:** Utilitzat per definir l'espai de noms (namespace) d'un element o d'un conjunt d'elements dins del document XML.
- **xmlns:xsi:** Defineix l'espai de noms XML Schema Instance i s'utilitza per a la validació d'esquemes XML.
- **xsi:schemaLocation:** Conté la ubicació de l'esquema XML utilitzat per validar el document XML.
- **xml:lang:** Especifica l'idioma principal del contingut de l'element.
- **xml:id:** S'utilitza per assignar un identificador únic a un element.
- **xml:base:** Especifica una URL base que es pot utilitzar per resoldre les URL relatives contingudes a l'element o el document XML.
- **xml:space:** Utilitzat per controlar l'espai en blanc (whitespace) a l'interior del contingut d'un element.
- **xml:version:** Indica la versió de XML utilitzada en el document. Per exemple, 1.0 o 1.1.
- **xml:encoding:** Indica la codificació de caràcters utilitzada en el document XML.
- **xml:standalone:** S'utilitza per especificar si el document XML està en mode "standalone" o si depèn d'entitats externes.
- **xml:comment:** Aquest no és un atribut estàndard, però s'utilitza per incloure comentaris dins del codi XML.

- Standalone: indica si el documento XML es independiente de otros recursos externos (yes/no), yes (el documento esta completo) no (el documento no se puede entender por si solo).

### ETIQUETAS

- Siempre van entre los símbolos "<" y ">".
- Son creadas por el autor y tienen que describir los datos que contienen.
- Dentro de una etiqueta podemos tener tantos atributos como necesitamos.

### ATRIBUTOS

Son elementos de metadatos que se pueden añadir a las etiquetas para proporcionar información adicional o controlar el comportamiento de cómo se procesa.

- Los atributos son parejas formadas por llave=valor.
- Tiene que ser un nombre XML valido y que no esté repetido.
- Podemos crear tantos atributos como necesitamos.
- Pueden estar entre comillas simples o dobles
- Todos los atributos deben tener alguna cosa asignada.

Se utilizan para añadir metadatos o control de procesamiento y presentación del documento XML.

- Metadatos: proporcionan información sobre un elemento.
- Control de contenido: influyen en como se procesa o se presenta el contenido de un elemento (ej. href: es una etiqueta enlace que puede especificar una URL).
- Idioma i codificación: especifica el idioma i la codificación de caracteres de un XML.
- Control de estilos: los atributos como "style" o "class" controlan el estilo CSS.

Atributos predefinidos:

•	<b>xmlns:</b> Utilitzat per definir l'espai de noms (namespace) d'un element o d'un conjunt d'elements dins del document XML.
•	<b>xmlns: xsi:</b> Defineix l'espai de noms XML Schema Instance i s'utilitza per a la validació d'esquemes XML.
•	<b>xsi:schemaLocation:</b> Conté la ubicació de l'esquema XML utilitzat per validar el document XML.
•	<b>xml:lang:</b> Especifica l'idioma principal del contingut de l'element.
•	<b>xml:id:</b> S'utilitza per assignar un identificador únic a un element.
•	<b>xml:base:</b> Especifica una URL base que es pot utilitzar per resoldre les URL relatives contingudes a l'element o el document XML.
•	<b>xml:space:</b> Utilitzat per controlar l'espai en blanc (whitespace) a l'interior del contingut d'un element.
•	<b>xml:version:</b> Indica la versió de XML utilitzada en el document. Per exemple, 1.0 o 1.1.
•	<b>xml:encoding:</b> Indica la codificació de caràcters utilitzada en el document XML.
•	<b>xml:standalone:</b> S'utilitza per especificar si el document XML està en mode "standalone" o si depèn d'entitats externes.
•	<b>xml:comment:</b> Aquest no és un atribut estàndard, però s'utilitza per incloure comentaris dins del codi XML.

## ENTIDADES

Las entidades en XML son mecanismos que permiten representar y referenciar caracteres especiales. Las mas comunes son:

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

## CDATA

- CDATA (Carácter de Datos), es una sección especial que permite incluir texto o datos con caracteres especiales.
- Es útil para añadir fragmentos de texto que no quieres que sea analizado o interpretado como parte de la estructura del XML.

```
<descripcio>
  <![CDATA[
    <p>Aquesta és una pàgina d'exemple.</p>
    <a href="http://www.exemple.com">Visita el nostre lloc web</a>
  ]]>
</descripcio>
```

El contenido dentro del bloque CDATA se trata como contenido y no como marcas del XML. En el ejemplo todo el contenido HTML se conserva sin ninguna interpretación, puede ser útil cuando queremos almacenar fragmentos de código con caracteres que podrían ser interpretados por el XML.



## COMENTARIOS

Se pueden añadir comentarios que serán ignorados por el programa que procese el documento.

```
<!-- Això és un comentari -->  
  
<!--  
I això també  
és un comentari  
-->
```

## INSTRUCCIONES DE PROCESO

Permiten que programas externos realicen tareas específicas. Se delimita por las marcas `<? ¿>` i el primer elemento dentro de ellas indica que programa tiene que procesar la instrucción.

```
<?nom_del_processor funcio(valor) ?>
```

## COMO ASOCIAR UN ARCHIVO CSS A UN DOCUMENTO XML

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

Esta instrucción es un ejemplo de como asociar un archivo CSS a un documento XML para aplicarle estilos.

## ESPACIOS DE NOMBRES (NAMESPACE)

Cada persona puede crear tantas etiquetas como quiera con el nombre que quiera. Por lo tanto nos podemos encontrar etiquetas repetidas al combinar documentos XML, que pueden significar cosas diferentes y crear conflictos.

Los espacios de nombres aseguran que los nombres de elementos y atributos sean únicos y no entren en conflicto.

Se declaran mediante el uso del atributo `xmlns` (XML Namespace)

La sintaxi bàsica seria:

```
<element_arrel xmlns="http://www.example.com/espai_de_noms">  
  <!-- Elements i atributs dins d'aquest espai de noms -->  
</element_arrel>
```

Nom	URL ESPAI DE NOMS
XML	http://www.w3.org/XML/1998/namespaces
XSL	http://www.w3.org/1999/XSL/Transform
XSL-FO	http://www.w3.org/1999/XSL/Format
XHTML	http://www.w3.org/1999/xhtml
SVG	http://www.w3.org/TR/svg

## ESQUEMAS Y VOCABULARIOS EN XML

- Los esquemas XML son como formularios que definen la estructura y el tipo de datos que se han de utilizar en el documento XML .
- El vocabulario XML son un conjunto de palabras y frases que tenemos que utilizar . Definen las etiquetas y elementos específicos que podemos utilizar en nuestro documento XML.

Se utilizan “lenguajes de definición de vocabulario” o “lenguajes de esquemas” (DTD).

- Un documento XML puede:
  - Estar bien formado, cumplir la sintaxis.
  - Ser valido, cumplir determinadas reglas y normas.
- Para establecer unas reglas de construcción en XML utilizamos DTD.
- Validar es un proceso habitual que realizaremos con programas especiales (procesadores o validadores).

## DTD

Un DTD es una especificación que define la estructura, los elementos y los atributos permitidos en documentos XML, proporcionando una guía para validar.

El DTD define:

- Tipo de elementos.
- Atributos.
- Entidades permitidas.
- También se pueden expresar restricciones.

Podemos crear DTDs de dos maneras:

- Ficheros externos: que se pueden compartir con diversos documentos.
- En el propio documento XML (inline)

## SINTAXIS BASICA

- **<!DOCTYPE:** indica que estem declarant un tipus de document DTD.
- **element:** especifica el nom de l'element que serà la arrel o element principal del document XML.
- **DTD identifier:** Aquest és un identificador que fa referència a la definició del tipus de document. Pot ser una ruta d'accés a un fitxer local o una URL que apunta a un fitxer DTD extern. Quan es fa referència a un fitxer extern, es parla de "Subconjunt Extern."
- **[ ]:** Els parèntesis quadrats indiquen que es pot incloure una llista opcional de declaracions d'entitats, que formen part del subconjunt intern del DTD.

## DTD INTERNO

El DTD se define dentro del propio documento XML.

```
<!DOCTYPE articles [  
    <!ELEMENT articles (article+)>  
    <!ELEMENT article (titol, autor, contingut)>  
    <!ELEMENT titol (#PCDATA)>  
    <!ELEMENT autor (#PCDATA)>  
    <!ELEMENT contingut (#PCDATA)>  
>]
```

## DTD EXTERNO

En este caso el DTD se define en un archivo separado y se asocia al documento XML (.dtd).

```
<!DOCTYPE articles SYSTEM "articles.dtd">
```

### articles.dtd

```
<!ELEMENT articles (article+)>  
<!ELEMENT article (titol, autor, contingut)>  
<!ELEMENT titol (#PCDATA)>  
<!ELEMENT autor (#PCDATA)>  
<!ELEMENT contingut (#PCDATA)>
```

También puede estar en una ubicación externa que puede ser:

```
<!DOCTYPE articles PUBLIC "https://exemple.com/dtd/articles.dtd">
```

- Privada, es la más común, la definimos nosotros y se utiliza con SYSTEM.
- Pública, la define un organismo de estandarización, se utiliza PUBLIC.

## ESTRUCTURA DTD

- Elementos
- Atributos
- Entidades
- Anotaciones

## DECLARACION DE ELEMENTOS

```
<!ELEMENT nom-element tipus-de-contingut>
```

- Nombre elemento: el elemento que se puede definir.
- Tipo de contenido: indica que tipo de contenido puede tener el elemento (PCDATA, EMPTY, ANY etc.).

## TIPOS DE CONTENIDOS

- Texto (#PCDATA) `<!ELEMENT cotxe (#PCDATA)>`
- EMPTY, no puede tener contenido. `<!ELEMENT linia_de_separacio EMPTY>`
- ANY, puede tener cualquier contenido `<!ELEMENT batiburrillo ANY>`
- Mixed, texto y otros elementos.
- Tipos de elementos, solo pueden tener subelementos. `<!ELEMENT article (titol, autor, contingut)>`

## MODELO DE CONTENIDOS

- Identificador general: `<!ELEMENT aviso (párrafo)>` aviso solo puede contener un solo párrafo.
- Secuencia: se genera mediante “,” `<!ELEMENT aviso (párrafo, titulo)>` aviso tiene que contener un párrafo seguido de un titulo.
- Opción: se utiliza la “|” en vez de la “,” para dar opción a una cosa o la otra. No hay limite de opciones.

## QUANTIFICADORES

(?)	= 0, 1 elemento
(*)	= 0 ó más elementos
(+)	= 1 ó más elementos
( )	= alternativa
(,)	= secuencia
EMPTY	= vacío
ANY	= cualquier estructura de subelementos
#PCDATA	= cadena de caracteres analizados

### Exemple Models de Continguts

```
<!ELEMENT aviso (titulo?, (parrafo+, grafico))>
```

- **<aviso>**: Aquest és l'element principal que estem definint.

La seua estructura és:

- **(titulo?, (parrafo+, grafico))**: Defineix l'estructura interna de l'element **<aviso>**.
  - **(titulo?, ...)**:
    - **<titulo>**: Este element és opcional (indicat per ?), es a dir, pot aparèixer **zero o una volta com a molt** dins de l'element **<aviso>**. No és obligatori tenir un **<titulo>** en cada **<aviso>**.
  - **(parrafo+, grafico)**: Esta part defineix que dins de l'element **<aviso>**, hi ha una seqüència de paràgrafs **<parrafo>** seguida d'un únic element **<grafico>**.

Les regles són les següents:

- **<parrafo+>**: Esta part indica que hi ha un o més elements **<parrafo>** (mínim un) dins de l'element **<aviso>**. Es a dir, hi ha almenys un paràgraf **<parrafo>**.
- **<grafico>**: Aquest element **<grafico>** ha de estar present després de la seqüència de paràgrafs **<parrafo>**. És a dir, hi haurà un únic element **<grafico>** després dels paràgrafs.

Això defineix una estructura flexible per a l'element **<aviso>**, on pots tenir zero o un **<titulo>**, seguit de paràgrafs (**<parrafo>**) i, finalment, un element **<grafico>**. També pots tenir múltiples paràgrafs abans del **<grafico>**. Aquesta declaració permet diversos formats de **<aviso>**, com ara:

- **<aviso>**
- **<aviso><titulo></titulo><parrafo></parrafo><parrafo></parrafo><grafico></grafico>**
- **<aviso><parrafo></parrafo><grafico></grafico>**

## ATRIBUTOS

Son elementos clave que se pueden utilizar para proporcionar información adicional.

- Van insertados en los elementos.
- Están formados por parejas nombre-valor.
- Los valores van entre comillas.
- Pueden ser obligatorios, opcionales, fijos...
- Aceptan valores de texto, números, listas de posibles valores etc.

`<f1 país="Países Baixos">Max Verstappen</f1>`

- Permiten añadir información adicional a los elementos de un documento.
- No pueden contener sub-atributos.
- Se utilizan para añadir información corta, sencilla y desestructurada.
- Solo se pueden especificar una vez.
- Pueden aparecer en cualquier orden.

## DECLARACION DE ATRIBUTOS EN DTD

`<!ATTLIST 1nombreElemento 2nombreAtributo 3tipoAtributo 4valorInicialAtributo>`

1. Identificador del elemento al que se aplica el atributo.
2. Nombre del atributo.
3. Tipo de atributo.
4. Valor por defecto del atributo.

### • Exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mensajes [
  <!ELEMENT mensajes (mensaje+)>
  <!ELEMENT mensaje (de, a, texto)>
  <!ATTLIST mensaje prioridad (normal|urgente) "normal">
  <!ELEMENT de (#PCDATA)>
  <!ELEMENT a (#PCDATA)>
  <!ELEMENT texto (#PCDATA)>
  <!ATTLIST texto idioma CDATA #REQUIRED>
]>
```

```
<mensajes>
  <mensaje prioridad="urgente">
    <de>Alfredo Regno</de>
    <a>Hans van Parijs</a>
    <texto idioma="holandés">
      Trobo Hans, hoe gaat het? ...
    </texto>
  </mensaje>
</mensajes>
```

- L'atribut `prioridad` pot estar en `<mensaje>` i pot tenir el valor `"normal"` o `"urgente"` sent `"normal"` el valor per defecte si no especifiquem l'atribut.
- Podem establir la necessitat o no d'especificar cada atribut.
- L'atribut `idioma`, pertany a `<texto>`, i pot contenir dades de caràcter (`CDATA`).
- `#REQUIRED`: indica que és obligatori especificar aquest atribut.

## TIPOS DE ATRIBUTOS

Típus d'Atribut	Descripció
<b>CDATA</b>	Permet qualsevol caràcter de dades (text).
<b>ENUMERAT</b>	Només permet valors específics.
<b>NMTOKEN</b>	Només permet noms de tokens (seqüències de caràcters alfanumèrics, guions baixos i altres caràcters especials, sense espais).
<b>NMTOKENS</b>	És similar a NMTOKEN, però permet una llista de noms de tokens separats per espais.
<b>ID</b>	Ha de ser un identificador únic per a l'atribut dins del document.
<b>IDREF</b>	Fa referència a l'ID d'un altre element.
<b>IDREFS</b>	És una llista de referències a IDs d'altres elements.
<b>ENTITAT</b>	Fa referència a un nom d'entitat definida al DTD.
<b>ENTITATS</b>	És una llista de noms d'entitats definides.

## VALORES DE ATRIBUTOS

- Los valores de los atributos pueden ser literales o referencias.
- En caso de literales los valores tienen que coincidir exactamente con los especificados en ENUMERATS o ser de tipo CDATA.
- Las referencias se utilizan para IDREF, IDREFS y ENTITAT

## VALORES PREDETERMINADOS DE ATRIBUTOS

- Puedes especificar un valor predeterminado para un atributo que se utilizara si no se proporciona un valor en el documento XML.
  - Valor entre comillas dobles o simples
- Los valores predeterminados pueden ser
  - #REQUIRED (obligatorio)
  - #IMPLIED (opcional)
  - #FIXED (tiene un valor fijo que no se puede cambiar)

## ENTIDADES

Se utilizan para definir los valores que se pueden reutilizar en todo el documento XML. Pueden ser símbolos, fragmentos de texto o incluso caracteres especiales. Sus principales usos son para ficheros, paginas web, imágenes etc.

<!ENTITY

## USO DE LAS ENTIDADES

Las entidades se utilizan poniendo & seguido del nombre de la entidad y luego un punto y coma.

```
<autor>&autor;</autor>  
<contingut>&llibre;</contingut>
```

## VENTAJAS DE LAS ENTIDADES

- Facilitan la reutilización de datos.
- Mejoran la legibilidad del documento XML.
- Permiten gestionar y mantener fácilmente el contenido que utilizan en diversos documentos.

## ENTIDADES INTERNAS

- Definidas directamente en el DTD del documento XML.
- Se utilizan principalmente para definir caracteres especiales o fragmentos de texto.
- Son siempre analizadas (una vez que se reemplaza la referencia a la entidad por su contenido pasa a ser parte del documento XML y se analiza por el procesador).

```
<!ENTITY nom "valor">
```

## ENTIDADES GENERALES EXTERNAS ANALIZADAS

Las entidades externas se definen en un fichero separado y se pueden utilizar en diferentes documentos XML.

```
<!ENTITY nom SYSTEM "URI o ruta de l'arxiu">
```

## ENTIDADES NO ANALIZADAS

- Si el contenido de la entidad es binario, no se tendría que interpretar como si fuera texto XML.
- Siempre son entidades generales y externas.

```
<!ENTITY logo SYSTEM "http://server.com/logo.gif">
```

- **Document DTD**

```
<?xml encoding="UTF-8"?>
```

→ Declaració XML que especifica la codificació del document com a UTF-8.

```
<!ELEMENT lista (persona)+>
```

→ Estableix que el teu document XML ha de començar amb un element **lista** que pot contenir un o més elements **persona**.

```
<!ELEMENT persona (nombre, email*, relacion?)>
```

→ Defineix l'element `persona`. Un element `persona` ha de contenir tres subelements: `nombre`, zero o més elements `email`, i zero o un element `relacion`.

```
<!ATTLIST persona id ID #REQUIRED>
```

→ Afegeix un atribut `id` a l'element `persona` que ha de ser de tipus ID i és requerit (és a dir, tota persona ha de tenir un `id` únic).

```
<!ATTLIST persona sexo (hombre | mujer) #IMPLIED>
```

→ Defineix l'atribut `sexo` a l'element `persona`. L'atribut `sexo` pot tenir els valors "hombre" o "mujer" i és opcional (marca `#IMPLIED`).

```
<!ELEMENT nombre (#PCDATA)>
```

→ Estableix que l'element `nombre` ha de contenir dades de text (PCDATA), és a dir, és un element de text.

```
<!ELEMENT email (#PCDATA)>
```

→ Estableix que l'element `email` també ha de contenir dades de text.

```
<!ELEMENT relacion EMPTY>
```

→ Defineix l'element `relacion` com buit (no pot contenir res més que atributs).

```
<!ATTLIST relacion amigo-de IDREFS #IMPLIED enemigo-de IDREFS #IMPLIED>
```

→ Afegeix dos atributs a l'element `relacion`: `amigo-de` i `enemigo-de`. Tots dos han de ser de tipus IDREFS (poden fer referència a IDs en altres elements) i són opcionals.



- **Document XML:**

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

→ Declara la versió de l'XML i la codificació que utilitza el document.

```
<!DOCTYPE lista SYSTEM "LISTA.dtd">
```

→ Enllaça el document XML amb la DTD anomenada "LISTA.dtd," que conté les regles de validació per al document XML.

```
<lista>
```

→ Obre l'element principal del document, 'lista'.

```
<persona sexo="hombre" id="alvaro">
```

→ Obre un element 'persona' i li afegeix dos atributs: 'sexo' amb el valor "hombre" i 'id' amb el valor "alvaro."

```
<nombre>Álvaro Álvarez</nombre>
```

```
<email>alvaroa@hotmail.com</email>
```

```
<relacion amigo-de="beatriz" />
```

→ Aquestes línies defineixen els subelements de l'element 'persona'. Tenim un 'nombre', un 'email' i un 'relacion' amb l'atribut 'amigo-de' que fa referència a "beatriz."

```
</persona>
```

→ Tanca l'element 'persona'.

```
<persona sexo="mujer" id="beatriz">
```

→ Defineix una altra persona anomenada Beatriz, amb un atribut 'sexo' igual a "mujer" i un atribut 'id' igual a "beatriz."

```
<nombre>Beatriz Bayo</nombre>
```

```
<email>bea@terra.com</email>
```

→ Aquestes línies defineixen el nom i l'adreça de correu electrònic de Beatriz.

```
</persona>
```

→ Finalment, aquesta línia tanca l'element 'persona', i així acaba el document XML.