

XML y DTD

INDICE

1. **Que es XML**
2. **Ventajas Y Desventajas**
3. **Herramientas Trabajo**
4. **Normas Básicas**
5. **Gramática**
6. **Partes**
7. **¿Bien Formados?**
8. **Estilos en XML**
9. **¿Qué es un DTD?**
10. **¿Para qué sirve?**
11. **Definición**
12. **Declaraciones**

¿QUÉ ES XML?

**ES UN METALENGUAJE
QUE PERMITE CREAR**

**LENGUAJES DE MARCAS PARA
ALMACENAR DATOS DE
FORMA LEGIBLE**

EJEMPLO ÁRBOL XML

```
<?xml version="1.0" encoding="UTF-8"?>
<empleados>
  ...
  <empleado>
    <nombre>Juan</nombre>
    <categoria>A</categoria>
    <antiguedad>10</antiguedad>
  </empleado>
  <empleado>
    <nombre>Ana</nombre>
    <categoria>B</categoria>
    <antiguedad>3</antiguedad>
  </empleado>
</empleados>
```

VENTAJAS Y DESVENTAJAS



Ventajas

**EXTENSIBLE Y COMPATIBLE
ENTRE PLATAFORMAS**

**FACILITA INTEROPERABILIDAD
ENTRE APLICACIONES**

**PERMITE TRANSFORMAR DATOS
EN INFORMACION ESTRUCTURADA**

**VERBOSIDAD EN ARCHIVOS
GRANDES**

**PARSING MÁS LENTO
COMPARADO CON JSON**

Desventajas



HERRAMIENTAS DE TRABAJO

"EDITORES Y
PROCESADORES
FACILITAN LA CREACIÓN,
EDICIÓN Y VALIDACIÓN
DE DOCUMENTOS XML."

[Inicio](#)

[Introducción](#)

[Contenido](#)

[Otros](#)

 **Sublime Text**



ALTOVA®
XMLSpy®
2025



BÁSICOS

Notepad++
SublimeText

AVANZADOS

XML Spy
Editix

NAVEGADORES
para visualización
básica



NORMAS BÁSICAS

1. TODAS LAS ETIQUETAS
DEBEN CERRARSE

2. SENSIBLE A MAYÚSCULAS
Y MINÚSCULAS

3. JERARQUÍA ESTRICTA
CON UNA ÚNICA RAÍZ

```
<?xml version="1.0" encoding="UTF-8"?>
<empleados>
  ...
  <empleado>
    <nombre>Juan</nombre>
    <categoria>A</categoria>
    <antiguedad>10</antiguedad>
  </empleado>
  <empleado>
    <nombre>Ana</nombre>
    <categoria>B</categoria>
    <antiguedad>3</antiguedad>
  </empleado>
</empleados>
```

**"La gramática
define las
reglas de
marcado
permitidas en
un lenguaje
basado en
XML."**

Gramática

PARTES DEL DOCUMENTO XHTML

ENCABEZADO

CUERPO

El prólogo

Lenguaje de Marcas > Ejercicio_2_XML_Alumnos.xml > Aula > Alumno

1

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

3

4 <Aula xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
5 xsi:noNamespaceSchemaLocation="Ejercicio_2_XML_Schema.xsd">

6 <Alumno dni="12345678A">

7 <Nombre>Juan</Nombre>

8 <Apellidos>Pérez López</Apellidos>

9 <NoRepetidor n_veces="0" />

10 <Sexo>hombre</Sexo>

11 <FechaNacimiento>

12 <Dia>15</Dia>

13 <Mes>5</Mes>

14 <Año>2000</Año>

15 </FechaNacimiento>

16 <Direccion codigoPostal="28001">

El cuerpo



¿QUE ES EL PROLOGO?

Es una declaración opcional que indica que ese documento esta basado en xml y le indica su versión y codificación

También indica el estilo a aplicar en el documento



EL CUERPO

El cuerpo es el resto del documento XML

No es opcional

Debe contener un solo elemento raíz y los propios datos del documento.



Documentos
bien formados

¿Que significa que un documento esté bien formado?

Un documento bien formado es aquel que cumple con las normas básicas de formato y puede ser analizado con un programa de análisis de XML

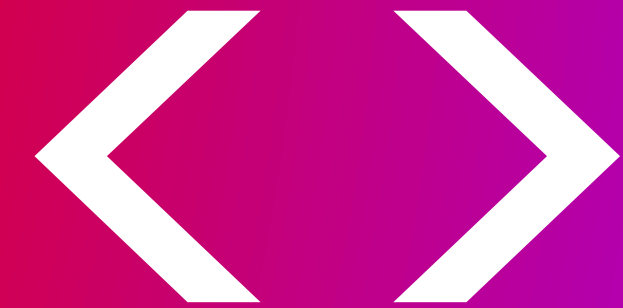
Un documento bien formado no es lo mismo que un documento válido

¿Entonces que es un documento válido?

Un documento válido es el cual está bien formado y además cumple la gramática especificada

Un documento puede estar bien formado pero puede ser no válido.

En cambio, un documento válido siempre va a ser un documento bien formado.



Estilos en XML

Aunque no es lo habitual, a un fichero XML se le puede aplicar una hoja de estilos

Existen dos tecnologías para aplicar estilos a un fichero XML

Un fichero XML está pensado para el intercambio de información y no la presentación de esta

CSS

XSLT

DTD(*DOCUMENT TYPE DEFINITION*)

¿Qué es un DTD?

Un DTD (*Document Type Definition*) es una descripción de elementos, estructura y anidamiento de un documento XML

```
return  
    <React.Fragment>  
      <div className="py-5">  
        <div className="container">  
          <Title name="our" title="product">  
            <div className="row">  
              <ProductConsumer>  
                {(value) => {  
                  console.log(value)  
                }}  
              </ProductConsumer>  
            </div>  
          </div>  
        </div>  
      </React.Fragment>
```

NO es un archivo XML

¿Para qué
sirve?



Tiene dos funciones

Normalizar una
estructura común
y mantener la
consistencia ente
varios archivos XML

Validar el archivo
XML si este
cumple con las
reglas definidas
en el DTD



¿Cómo lo
definimos?

Existen dos formas

INTERNAMENTE

Dentro del propio archivo XML

EXTERNAMENTE

Enlazando un archivo .dtd externo

INTERNAMENTE



Lo definimos en la
cabecera del archivo
XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejemplo [
    <!ELEMENT ejemplo (nombre, apellido, edad)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT apellido (#PCDATA)>
    <!ELEMENT edad (#PCDATA)>
]>
<ejemplo>
    <nombre>Nombre</nombre>
    <apellido>Apellido</apellido>
    <edad>Edad</edad>
</ejemplo>
```



EXTERNAMENTE

Enlazamos a nuestro
XML un archivo .dtd
externo

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejemplo SYSTEM "ejemplo.dtd">
<ejemplo>
    <nombre>Nombre</nombre>
    <apellido>Apellido</apellido>
    <edad>Edad</edad>
</ejemplo>
```

```
<!ELEMENT ejemplo (nombre, apellido, edad)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT edad (#PCDATA)>
```

ejemplo.dtd



Carácter



Privados

Son los definidos por nosotros mismos y se identifican con la palabra SYSTEM

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

Públicos



Son los definidos por una organización de estándares y se identifican con la palabra PUBLIC

```
<!DOCTYPE ejemplo PUBLIC "public-id" "ejemplo.dtd">
```

Declaraciones



Elementos

Entidades

Atributos

Notaciones

Elementos



Son las etiquetas
que constituyen
nuestro XML

Pueden o no tener
contenido


Pueden contener:

- Otros elementos
- Texto
- Ambas cosas

Ejemplos

```
<!DOCTYPE ejemplo[
  <!ELEMENT ejemplo (nombre, apellido, edad, hijos)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido (#PCDATA)>
  <!ELEMENT edad (#PCDATA)>
  <!ELEMENT hijos EMPTY>
]>
<ejemplo>
  <nombre>Nombre</nombre>
  <apellido>Apellido</apellido>
  <edad>Edad</edad>
  <hijos/>
</ejemplo>
```

```
<!DOCTYPE ejemplo[
  <!ELEMENT ejemplo (nombre, apellido, edad, hijos, direccion*)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido (#PCDATA)>
  <!ELEMENT edad (#PCDATA)>
  <!ELEMENT hijos EMPTY>
  <!ELEMENT direccion (#PCDATA)>
]>
<ejemplo>
  <nombre>Nombre</nombre>
  <apellido>Apellido</apellido>
  <edad>Edad</edad>
  <hijos/>
  <direccion>Villa ejemplo, 7</direccion>
  <direccion>Villa ejemplo, 8</direccion>
</ejemplo>
```



Ocurrencias



"nombre_hijo"

"nombre_hijo*"

"nombre_hijo+"

"nombre_hijo?"



Contenido

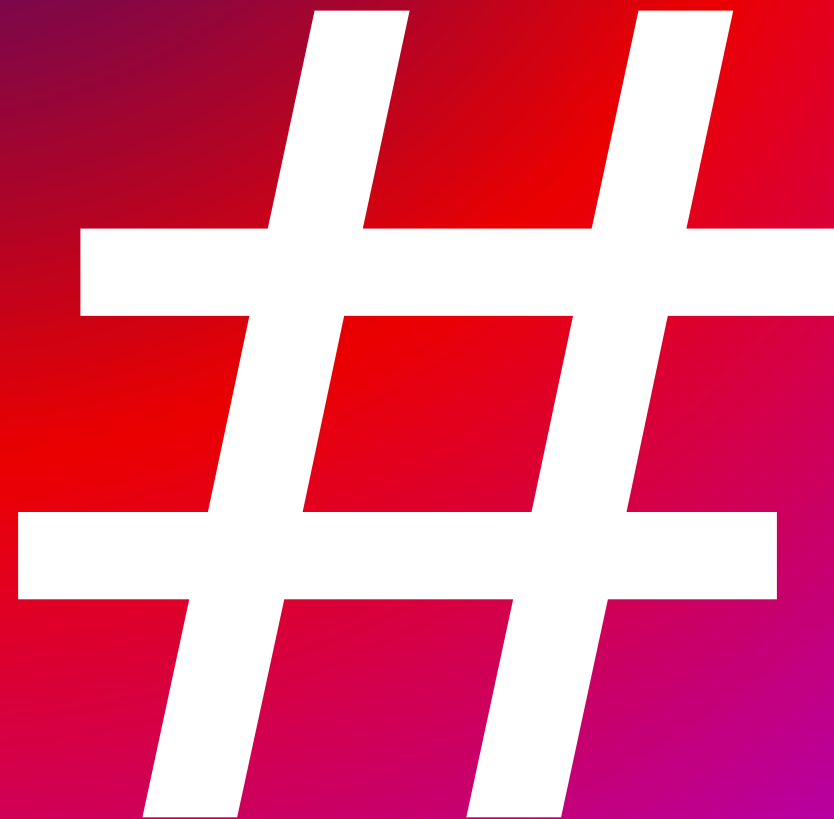
#PCDATA

EMPTY

ANY

MIXTO

Atributos



Son los valores que definen las propiedades de un elemento

Se declaran mediante ATTLIST

```
<!DOCTYPE ejemplo[
  <!ELEMENT ejemplo (nombre, apellido, edad, hijos, familia)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido (#PCDATA)>
  <!ELEMENT edad (#PCDATA)>
  <!ELEMENT hijos EMPTY>
  <!ELEMENT familia EMPTY>
  <!-- ATTLIST familia apellidos CDATA #REQUIRED -->
]>
<ejemplo>
  <nombre>Nombre</nombre>
  <apellido>Apellido</apellido>
  <edad>Edad</edad>
  <hijos/>
  <familia apellidos="Ejemplo Ejemplo"></familia>
</ejemplo>
```




Tipos y valores

Type	Description
CDATA	The value is character data
(<i>en1 en2 ..</i>)	The value must be one from an enumerated list
ID	The value is a unique id
IDREF	The value is the id of another element
IDREFS	The value is a list of other ids
NMTOKEN	The value is a valid XML name
NMTOKENS	The value is a list of valid XML names
ENTITY	The value is an entity
ENTITIES	The value is a list of entities
NOTATION	The value is a name of a notation
xml:	The value is a predefined xml value

Value	Explanation
<i>value</i>	The default value of the attribute
#REQUIRED	The attribute is required
#IMPLIED	The attribute is optional
#FIXED <i>value</i>	The attribute value is fixed

Es **obligatorio** poner el valor
que van a tomar los
atributos

Entidades



Se usan para definir
atajos en
documentos XML

Se declaran
mediante ENTITY

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE ejemplo[
    <!ELEMENT ejemplo (nombre, apellido, edad)>
    <!ELEMENT nombre (#PCDATA)>
    <!ELEMENT apellido (#PCDATA)>
    <!ELEMENT edad (#PCDATA)>
    <!-- ENTITY nombre "Nombre" -->
]>
<ejemplo>
    <nombre>&nombre;</nombre>
    <apellido>Apellido</apellido>
    <edad>Edad</edad>
</ejemplo>
```

[Inicio](#)

[Introducción](#)

[Contenido](#)

[Otros](#)

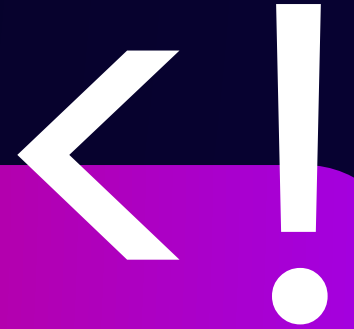
&

Tipos

Internas

Son las que definen un valor
para ese archivo XML

```
<!ENTITY estadoCivil "Casado">
```



Externas

Hacen referencia a un
archivo externo

```
<!ENTITY estadoCivil SYSTEM "ejemplo.dtd">
```


Paramétricas



Funcionan igual
que las
generales, pero
se sustituyen en
todo el
documento

```
<!ENTITY % p "(#PCDATA)">  
<!ELEMENT ejemplo (nombre, apellido, edad)>  
<!ELEMENT nombre %p;>  
<!ELEMENT apellido (#PCDATA)>  
<!ELEMENT edad %p;>
```

Notaciones



Permiten indicar el formato de los datos que no se encuentran en formato XML

Se declaran mediante
NOTATION

```
<!NOTATION gif SYSTEM "imagen.gif">  
<!NOTATION jpg SYSTEM "imagen.jpg">
```

Espacios de nombre



Se usan cuando definimos varias regulaciones para un mismo archivo XML

Se declaran mediante `xmlns:prefijo`

```
<?xml version="1.0"?>
<cli:cliente xmlns:cli='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/cliente'
             xmlns:ped='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/pedido'>
  <cli:numero_ID>1232654</cli:numero_ID>
  <cli:nombre>Fulanito de Tal</cli:nombre>
  <cli:telefono>999999999</cli:telefono>
  <ped:pedido>
    <ped:numero_ID>6523213</ped:numero_ID>
    <ped:articulo>Caja de herramientas</ped:articulo>
    <ped:precio>187,90</ped:precio>
  </ped:pedido>
</cli:cliente>
```

```
<?xml version="1.0"?>
<cli:cliente xmlns:cli='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/cliente'
  ..
  xmlns:ped='http://es.wikipedia.org/wiki/Espacio_de_nombres_XML/pedido'>
  <cli:numero_ID>1232654</cli:numero_ID>
  <cli:nombre>Fulanito de Tal</cli:nombre>
  <cli:telefono>99999999</cli:telefono>
  <ped:pedido>
    <ped:numero_ID>6523213</ped:numero_ID>
    <ped:articulo>Caja de herramientas</ped:articulo>
    <ped:precio>187,90</ped:precio>
  </ped:pedido>
</cli:cliente>
```

[Inicio](#)

[Introducción](#)

[Contenido](#)

[Otros](#)

Gracias