

● Contenidos:

[Introducción.](#)

[1. Asociación de un DTD a documento XML.](#)

[2. Definiciones en un DTD:](#)

[A. Elementos o etiquetas.](#)

[B. Atributos.](#)

[C. Entidades.](#)

[D. Notaciones.](#)

[E. Secciones Condicionales.](#)

[F. Comentarios.](#)

[3. Generación automática de DTD.](#)

[4. Limitaciones de los DTD.](#)

[5. Verificación de un documento XML: Bien Formado y Validación.](#)

La DTD o documento de definición de tipos indica el formato que debe tener el documento XML (estructura y sintaxis).

El DTD se encarga de **definir**:

- **Léxico**: conjunto de palabras que se pueden utilizar.
- **Sintaxis**: conjunto de reglas que establecen la estructura del documento.

Nota: recordar cabecera documento XML:

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

Si deseamos que el navegador valide el documento:

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

1. Asociación de un DTD a documento XML.



La asociación puede realizarse de forma:

- Interna o Externa al documento XML.
- Pública (PUBLIC) o Privada (SYSTEM) al equipo que validará el documento XML.

Para asociar un documento XML a su validación siempre se incluirá en el documento XML:

```
<!DOCTYPE raíz ... >
```

Donde “raíz”: elemento raíz del documento XML.

- Declaración del DTD: definirá la forma de asociación del DTD al documento XML.

Los documentos DTD externos tendrán extensión .dtd

Tipo DTD	Sintaxis
Privado e interno	<!DOCTYPE raíz [reglas]>
Privado y externo	<!DOCTYPE raíz SYSTEM URL>
Privado y mixto	<!DOCTYPE raíz SYSTEM URL [reglas]>
Público e interno	No tiene sentido (*)
Público y externo	<!DOCTYPE raíz PUBLIC FPI URL>
Público y mixto	<!DOCTYPE raíz PUBLIC FPI URL [reglas]>

(*) La combinación Público e interno no tiene sentido, ya que si lo almacenamos en nuestro sistema no puede estar accesible para otras personas o aplicaciones.

- a) **En el propio documento** (Interna y, lógicamente privada, ya que va en el documento XML):

```
<!DOCTYPE raíz [...códigoDTD...]>
```

· [...códigoDTD...]: dentro de los símbolos [...] se especifican las instrucciones DTD.

- b) **En un documento Externo Privado:**

```
<!DOCTYPE raíz SYSTEM "ruta_URL_al_DTD">
```

Las reglas aparecen en un archivo independiente, en el propio equipo.

- c) **En un documento Externo Público:**

```
<!DOCTYPE raíz PUBLIC "nombreDTD" "DTD_URL">
```

Las reglas aparecen en un archivo independiente, en equipo o contenedor público.

· “nombreDTD”: es el nombre público que se le da al DTD en cuestión (uso de FPI).

Si disponemos de un repositorio de DTDs públicos (como ocurre en entornos de trabajo como Oxygene por ejemplo) le cargaría sin ir a Internet.

· “DTD_URL”: si el “nombreDTD” no es reconocido se usa la dirección URL para descargarlo y utilizarlo.

Formato del FPI: Está compuesto de **4 campos separados por //**.

1) **Tipo de norma:**

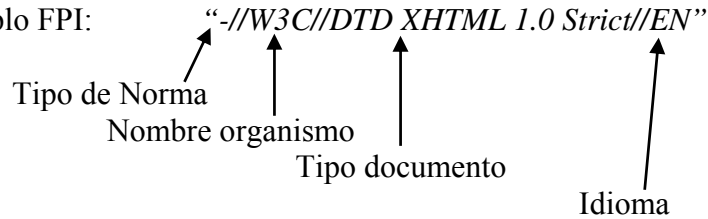
- No aprobada por una norma formal (-).
- Aprobada por organismo no oficial (+).
- Aprobada por organismo oficial (referencia al estándar).

2) **Nombre del organismo responsable del estándar.**

3) **Tipo de documento**, suele incluir la versión.

4) **Idioma.**

Ejemplo FPI:



URN en lugar de FPI:

Cada vez está más extendido el uso de un URN (Uniform Resource Name) en lugar del FPI como identificador.

Para crear una URN a partir de un FPI simplemente añadimos al principio "urn:publicid:" y a continuación el FPI sustituyendo // por : y los espacios en blanco por +.

Ejemplo:

`"-//W3C//DTD XHTML 1.0 Transitional//EN"`

Convertido a URN sería:

`"urn:publicid:-:W3C:DTD+XHTML+1.0+Transitional:EN"`

2. Definiciones en un DTD.

En un código DTD (tanto externo como interno) se pueden definir 4 tipos de componentes:

- Los **elementos** que se pueden utilizar en un documento XML. En esta definición se indica además que pueden contener dichos elementos.
- Los **atributos** que pueden poseer los elementos. Además, incluso indicando sus posibles valores válidos.
- Las **entidades** que puede utilizar el documento XML.
- Las **notaciones**.
- **Secciones condicionales**.

A. Elementos o etiquetas:

El primer elemento será el nodo raíz, que deberá ser único, con una única ocurrencia (no se puede repetir).

`<!ELEMENT nombre tipocontenido >`

Donde:

- nombre: es el identificador que tendrá el elemento en el documento XML (hay que recordar que se distingue entre mayúsculas y minúsculas, y no se deben usar símbolos no permitidos).
- tipocontenido: indica el funcionamiento del elemento, relativo al contenido que puede tener.

○ Tipos de contenido en los elementos:

- **EMPTY**: el elemento no podrá tener contenido alguno, es un elemento vacío. Pueden tener atributos.
- **ANY**: permite cualquier contenido en el elemento.
- **(Contenido concreto)**: tiene que aparecer un contenido concreto. Dicho contenido se indica entre paréntesis, pudiendo ser:
 - **(#PCDATA)**: Datos.
 - **(lista_elementos)**: Descendientes. Se puede establecer:
 - Secuencia:
 - (elemento,elemento): indica el orden en que deben aparecer los elementos (,).
 - (elemento|elemento): indica que los elementos son excluyentes (|), es decir, aparecerá uno o el otro, no los dos a la vez.
 - Cardinalidad (¿,*,+): indica el número de veces que puede aparecer un elemento o una secuencia de ellos.
 - ?: contenido opcional. Puede aparecer una sola vez o no aparecer. [0-1]
 - *: contenido opcional y repetible. Puede no aparecer, o aparecer varias veces. [0-N]
 - +: contenido obligatorio y repetible. Tiene que aparecer, e incluso aparecer varias veces. [1-N]

Si un elemento tiene descendientes, después de enumerarlos hay que describirlos.

- **(#PCDATA, lista_elementos)**: Mixto. Datos y descendientes. (PCDATA deberá aparecer el primer elemento de la lista)
- **((lista_elementos)|(lista_elementos))**: Combinaciones de definiciones. Establece que puede aparecer una lista u otra lista.

B. Atributos:

Los atributos permiten añadir información a un elemento.

```
<!ATTLIST elemento
    nombre_atributo tipo_atributo carácter valorPorDefecto
    nombre_atributo tipo_atributo carácter valorPorDefecto
    ...
>
```

Donde:

- **elemento**: es el nombre del elemento que podrá utilizar el atributo.
- **nombre_atributo**: es el identificador del atributo que estamos declarando (y que debe de cumplir las reglas de identificadores de XML).
- **tipo_atributo**: es el tipo de valores que podemos asignar al atributo.
- **carácter**: indica las características de los valores que puede tomar el atributo: si es obligatorio, si hay valor por defecto, etc.
- **valorPorDefecto**: permite especificar un valor que el atributo tomará si no especifica otro explícitamente en el documento XML.

○ Tipo de Atributo:

- **CDATA**: Texto.
- **(valor1 | valor2 | ...)**: Enumeración. Lista de valores de entre los cuales el atributo deberá tomar uno.
- **ID**: identificador que permite identificar, diferenciar, a un elemento de forma única.
Los atributos ID solo pueden indicar #IMPLIED o #REQUIRED en el apartado del valor por defecto.
- **IDREF**: representa el valor de un atributo ID de otro elemento. Mediante el uso de ID e IDREF se pueden relacionar elementos.
- **IDREFS**: múltiples IDs de otros elementos separados por espacios.
- **NMTOKEN**: el valor del atributo será un texto sin espacios en su interior ni símbolos reservados.
(sólo puede contener letras, dígitos, punto [.], guión [-], subrayado [_] y dos puntos [:])
- **NMTOKENS**: el valor del atributo puede contener varios valores de tipo NMTOKEN separados por espacios.
- **ENTITY**: nombre de una entidad (debe estar declarada en el DTD).
- **ENTITIES**: lista de nombres de entidades separadas por espacios (declaradas en el DTD).
- **NOTATION**: nombre de una notación (debe estar declarada en el DTD).

○ Carácter:

- **"valor"**: establece un valor por defecto para el atributo.
- **#IMPLIED**: el atributo es opcional y no se le asigna ningún valor por defecto.
- **#REQUIRED**: el atributo es obligatorio, pero no se le asigna un valor por defecto.
- **#FIXED "valor"**: el atributo es obligatorio y su valor no puede ser otro más que el establecido.

C. Entidades:

Se utilizan para definir referencias a valores u objetos (ficheros, sitios web, imágenes, etc.), de manera que una vez definida una entidad podemos sustituir su valor por un nombre.

Existen 3 **tipos de entidades**:

- Entidades generales (internas o externas)
- Entidades parámetro (internas o externas)
- Entidades no procesadas (unparsed)

a) Entidades Generales Internas: definir abreviaturas que serán sustituidas por su valor pasando a formar parte del XML y por tanto siendo procesadas.

`<!ENTITY nombre_entidad "valor_entidad">`

Para usar en un documento XML la entidad declarada:

`&nombre_entidad;`

b) Entidades Generales Externas: en este caso el valor se obtiene de un archivo de texto.

`<!ENTITY nombre_entidad privacidad "url_archivo">`

Para usar en un documento XML la entidad declarada:

`&nombre_entidad;`

c) Entidades Parámetro: se utilizan para agrupar elementos del DTD que se repiten con cierta frecuencia.

Para referenciarlos se utiliza `%` en lugar de `&`.

• Entidades parámetro internas:

`<!ENTITY % nombre_entidad "valor_entidad">`

• Entidades parámetro externas:

`<!ENTITY % nombre_entidad privacidad "url">`

Entidades No Procesadas: entidades generales externas cuando el archivo no contiene texto sino datos binarios: imágenes, videos, ejecutables, etc.

En estos casos el contenido del archivo no debe ser procesado como si fuera texto XML.

`<!ENTITY entidad privacidad "url" NDATA notacion>`

d) Entidades Definidas en XML:

En XML están definidas las siguientes entidades, de forma implícita, por lo que no hay que declararlas en ningún DTD (todos los analizadores de XML estándar conocen estas entidades).

Entidad	Significado
<code>&lt;</code>	Símbolo de menor (<)
<code>&gt;</code>	Símbolo de mayor (>)
<code>&amp;</code>	Ampersand (&)
<code>&apos;</code>	Comilla simple (')
<code>&quot;</code>	Comilla doble (")

e) Entidades para referencias a Caracteres Especiales:

Para indicar un carácter especial que no está contenido en nuestro teclado

- Código del carácter en decimal: `&#código;`
- Código del carácter en hexadecimal: `&#xnúmero;`

D. Notaciones:

Se pueden utilizar para especificar el formato de entidades externas (datos no XML). Dichas entidades externas no las analizará un procesador XML, sino que serán tratadas por el programa que procese el documento.

La declaración de la notación sería:

```
<!NOTATION nombre_notación privacidad "identificador">
```

Las notaciones podrán ser usadas en Entidades o en Atributos:

La declaración de la entidad que la utiliza: (entidades no procesadas)

```
<!ENTITY entidad privacidad "url" NDATA notacion>
```

La declaración del atributo que la utiliza:

```
<!ATTLIST nombre_elemento nombre_atributo NOTATION valor_defecto>
```

E. Secciones Condicionales:

En DTD externas se pueden definir las secciones IGNORE e INCLUDE, para ignorar o incluir declaraciones.

Reglas ignoradas:

```
<![ IGNORE [ declaraciones ]]>
```

Reglas incluidas:

```
<![ INCLUDE [ declaraciones ]]>
```

F. Comentarios:

```
<!-- comentarios -->
```

3. Generación automática de DTD

Existen herramientas que permiten generar de forma automática un DTD a partir de un documento XML.

Hay que tener en cuenta que estos DTDs generados automáticamente suponen una gran ayuda y son un buen punto de partida, pero que será necesario refinarlos para obtener el DTD correcto.

4. Limitaciones de los DTD

- Un DTD no es un documento XML y no se puede verificar si está bien formado (sólo si es válido, pero no si está bien formado).
- No se pueden fijar restricciones sobre los valores de elementos y atributos, como tipo de dato, tamaño...
- No soportan espacios de nombres.
- Sólo se pueden enumerar los valores de los atributos, no de los elementos.
- Sólo se pueden establecer valores por defecto para los atributos, no para los elementos.
- El control de la cardinalidad (número de repeticiones) de los elementos es limitado.

5. Verificación de un documento XML: Bien Formado y Validación

- Podemos usar el validador del W3C, que se encuentra en la página:
http://validator.w3.org/#validate_by_uri.
- La mayoría de los navegadores no tienen en cuenta si el documento se ajusta a su DTD sino simplemente si el documento está bien formado. Es decir, sólo tienen en cuenta que el documento se ajuste a las normas básicas de creación de archivos XML. Sin embargo, podemos poner el atributo standalone="yes" en la primera declaración del archivo XML para comprobar si el XML se ajusta a la DTD:

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

- Verificación a través de la aplicación XML Copy Editor: