

2.VALIDACIÓN DE DATOS -DTD

2.1 DOCUMENTO VÁLIDO

La DTD o gramática de un documento XML es la que define el número y el orden de aparición de los elementos, los atributos y sus tipos. Es decir la DTD declara las normas para que un documento XML sea válido.

Ejemplo de un documento XML con una DTD dentro:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>

<!--Declaraciones de la DTD-->
<!DOCTYPE nota [
<!ELEMENT nota (para,de,cabecera,mensaje)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecera (#PCDATA)>
<!ELEMENT mensaje (#PCDATA)>
]>
<nota>
<para>Pedro</para>
<de>Jose</de>
<cabecera>Recordatorio</cabecera>
<mensaje>No te olvides de echar la primitiva</mensaje>
</nota>
```

Lo mismo pero codificando la DTD aparte, dos documentos, uno con extensión dtd, nota.dtd:

```
<!ELEMENT nota (para,de,cabecera,mensaje)>
<!ELEMENT para (#PCDATA)>
<!ELEMENT de (#PCDATA)>
<!ELEMENT cabecera (#PCDATA)>
<!ELEMENT mensaje (#PCDATA)>
```

otro nota.html desde el que se llama a la DTD anterior:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>

<!DOCTYPE nota SYSTEM "nota.dtd">
<nota>
<para>Pedro</para>
<de>Jose</de>
<cabecera>Recordatorio</cabecera>
<mensaje>No te olvides de echar la primitiva</mensaje>
</nota>
```

Para validar nos descargamos el programa " xml copy editor", de sourceforge.net:

<http://sourceforge.net/projects/xml-copy-editor/>

También se puede validar usando otros programas como XmlSpy (de AltovA).

Para la validación online: <http://www.xmlvalidation.com>

Los navegadores no validan contra la DTD.

Es preferible usar DTD externas, así podemos intercambiar datos con otras personas siguiendo las mismas reglas.

2.2. ELEMENTOS

Son los nodos del árbol de un documento XML. Se declaran con la etiqueta <ELEMENT>

Elementos vacíos

Prohíbe que un elemento tenga hijos o datos textuales.

<!ELEMENT nombre-elemento EMPTY>

Ejemplo:

<!ELEMENT br EMPTY>

Ejemplo XML:

Elementos con datos textuales

<!ELEMENT nombre-elemento (#PCDATA)>

Ejemplo:

<!ELEMENT de (#PCDATA)>

Ejemplo XML:

<de>Pedro</de>

Elementos con hijos (secuencias)

Elementos con uno o más hijos son declarados con el nombre de los elementos hijos entre paréntesis:

<!ELEMENT nombre-elemento (child1)>
or

<!ELEMENT nombre-elemento (child1,child2,...)>

Ejemplo:

<!ELEMENT nota (para,de,cabecera,mensaje)>

Cuando los hijos son declarados en una secuencia separada por comas, los hijos deben aparecer en la misma secuencia en el documento. Los hijos tienen que ser declarados después y pueden tener a su vez más hijos.

Ejemplo:

<!ELEMENT nota (para,de,cabecera,mensaje)>

<!ELEMENT para (#PCDATA)>

<!ELEMENT de (#PCDATA)>

<!ELEMENT cabecera (#PCDATA)>

<!ELEMENT mensaje (#PCDATA)>

Declarar Sólo una Ocurrencia de un elemento

<!ELEMENT nombre-elemento (child-name)>

Ejemplo:

<!ELEMENT nota (mensaje)>

El elemento mensaje debe aparecer una sola vez dentro del elemento nota.

Declarar una ocurrencia como mínimo

<!ELEMENT nombre-elemento (child-name+)>

Ejemplo:

<!ELEMENT nota (mensaje+)>

El elemento mensaje debe aparecer una o más veces dentro del elemento nota.

Declarar cero o más ocurrencias de un Elemento

<!ELEMENT nombre-elemento (child-name*)>

Ejemplo:

<!ELEMENT nota (mensaje*)>

El elemento mensaje puede no aparecer o aparecer varias veces dentro del elemento nota.

Declarar cero o una ocurrencia de un elemento

<!ELEMENT nombre-elemento (child-name?)>

Ejemplo:

<!ELEMENT nota (mensaje?)>

El elemento mensaje puede aparecer una vez o ninguna dentro del elemento nota.

Declarando un contenido u otro

Ejemplo:

```
<!ELEMENT nota (para,de,cabecera,(mensaje|cuerpo))>
```

El elemento nota debe contener un elemento para, otro elemento de, otro elemento cabecera y luego un elemento mensaje o un elemento cuerpo.

Mezclando datos textuales y elementos

Ejemplo:

```
<!ELEMENT nota (#PCDATA|para|de|cabecera|mensaje)*>
```

El elemento nota puede contener cero o más ocurrencias de texto o elemento para, o de, o cabecera, o mensaje .

2.3. ATRIBUTOS

Siempre que sea posible es mejor usar elementos que atributos.

Un atributo se declara así:

```
<!ATTLIST nombre-del-elemento nombre-de-atributo tipo-del-atributo valor-por-defecto>
```

DTD ejemplo:

```
<!ATTLIST pago modo CDATA "cheque">
```

XML ejemplo:

```
<pago modo="cheque" />
```

El tipo del atributo puede ser:

Tipo	Descripción
CDATA	Una cadena
(en1 en2 ..)	Un valor de una lista
ID	Un único id
IDREF	El id de otro elemento
IDREFS	Una lista de otros id

ENTITY	Una entidad
ENTITIES	Una lista de entidades

El valor por defecto puede ser uno de los siguientes:

Valor	Explicación
<i>valor</i>	Valor por defecto
#REQUIRED	El atributo es obligatorio
#IMPLIED	El atributo no es obligatorio
#FIXED <i>valor</i>	El atributo tiene un valor fijo

VALOR POR DEFECTO

DTD:

```
<!ELEMENT cuadrado EMPTY>
```

```
<!ATTLIST cuadrado ancho CDATA "0">
```

Valido XML:

```
<cuadrado ancho="100" />
```

Si el ancho no se especifica, su valor por defecto es 0.

#REQUIRED

DTD:

```
<!ATTLIST persona numero CDATA #REQUIRED>
```

Valida XML:

```
<persona numero="5677" />
```

Invalida XML:

```
<persona />
```

Si usamos #REQUIRED el atributo no tiene valor por defecto y forzamos a que el atributo aparezca.

#IMPLIED

DTD:

```
<!ATTLIST contacto fax CDATA #IMPLIED>
```

Valid XML:

```
<contacto fax="555-667788" />
```

Valid XML:

```
<contact />
```

Si usamos #IMPLIED el atributo no tiene valor por defecto y no obligamos a escribir el atributo.

#FIXED

DTD:

```
<!ATTLIST emisor company CDATA #FIXED "Microsoft">
```

Valida XML:

```
<emisor company="Microsoft" />
```

Invalida XML:

```
<emisor company="W3Schools" />
```

Usamos #FIXED cuando queremos que el atributo tenga un valor fijo, no se admite otro valor.

ENUMERACIONES

DTD:

```
<!ATTLIST pago modo (check|cash) >
```

XML ejemplo:

```
<pago modo="check" />
```

o

```
<pago modo="cash" />
```

Se usan cuando el valor del atributo es uno de un conjunto de valores fijos.

ATRIBUTOS DE TIPO ID O IDREF

El atributo puede utilizarse como identificador de un elemento, este identificador debe ser único en todo

el documento.

Los atributos de tipo IDREF son referencias a los identificadores. El valor del atributo IDREF debe corresponder a un identificador de elemento existente.

Utilizamos IDREFS para especificar varias referencias a identificadores dentro de la declaración de un atributo.

Ejemplo:

DTD:

```
<!ELEMENT librococina (receta +,ingrediente+)>
```

```
<!ELEMENT receta #PCDATA>
```

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

```
<!ELEMENT ingredient e #PCDATA>
```

```
<!ATLIST ingredient ref IDREFS #IMPLIED>
```

XML válidos:

```
<librococina>
```

```
<receta id="rec_1">Mousse de chocolate</receta>
```

```
<receta id="rec_2">Flan</receta>
```

```
<receta id="rec_3">Fondue de chocolate</receta>
```

```
<ingrediente ref="rec_1 rec_3">chocolate</ingrediente>
```

```
<ingrediente ref="rec_2">vainilla</ingrediente>
```

```
</librococina>
```

2.4. ENTIDADES

Permiten definir constantes en los documentos XML.

DECLARACIÓN INTERNA

Sintaxis:

```
<!ENTITY nombre-entidad "valor">
```

DTD:

<!ENTITY escritor "Ken Follet">

XML:

<autor>&escritor;</autor>

Una entidad tiene 3 partes: &, nombre de la entidad y ;

DECLARACIÓN EXTERNA

Sintaxis:

<!ENTITY nombre-entidad SYSTEM "URL/URI">

DTD

<!ENTITY documento SYSTEM "http://localhost/docxml/otrodoc.xml">

XML

<referencia>&documento;</referencia>

Las referencias a la entidad externa serán reemplazadas automáticamente por el contenido del documento al que se hace referencia.

ENTIDADES PREDEFINIDAS

Existen una serie de entidades ya definidas

Entidad Valor

< <

> >

& &

' ‘

" “

 espacio en blanco

Se puede escribir cualquier carácter usando su código hexadecimal: ©

2.5. VALIDACIÓN WEB

La validación es el proceso que asegura que un documento escrito en un determinado lenguaje (por ejemplo XHTML) cumple con las normas y restricciones de ese lenguaje. Las normas y restricciones

de los documentos escritos en XML (y en sus lenguajes derivados, como XHTML) se definen en el DTD o *Document Type Definition* ("*Definición del Tipo de Documento*").

El concepto de validación es objeto de controversia en el ámbito del diseño web. Por una parte, la validación no es obligatoria y las páginas web se pueden ver bien sin que sean válidas. Por otra parte, una página válida es más correcta que otra página que no lo sea, ya que cumple con las normas y restricciones impuestas por XHTML.

Debido a esta controversia, algunos diseñadores consideran que se da demasiada importancia a la validación de las páginas y a la creación de páginas válidas. El resto de diseñadores argumentan que si la especificación de XHTML impone una serie de normas y restricciones, lo más correcto es que las páginas web las cumplan, aunque no sea obligatorio.

En cualquier caso, el proceso de validación consiste en probar página a página si su código HTML pasa la prueba de validación. Los validadores son las herramientas que se utilizan para validar cada página. Algunos editores de páginas web, como dreamweaver incluyen sus propios validadores y el organismo W3C ha creado una herramienta gratuita para la validación de las páginas:

<http://validator.w3.org/>