

# Introducción ó uso de métodos de definición de documentos XML

## Sumario

- 1 Introducción
- 2 ¿Que é un Schema XML?
- 3 Os Schemas XML son os sucesores dos DTD
- 4 Os Schemas XML soportan tipos de datos
- 5 Os Schemas XML usan sintaxe XML
- 6 Os Schemas XML garanten a comunicación segura de datos
- 7 Ben formado non é suficiente

## Introdución

Un DTD (Document Type Definition) é a descrición da estrutura e sintaxe dun documento XML ou SGML. Úsase para validar a estrutura de múltiples documentos e manter a súa consistencia.

En moitos aspectos os DTD considéranse anticuados e úsanse cada vez menos, grazas sobre todo á existencia de linguaxes de validación mais modernos e máis axeitadas para validar os documentos XML.

Os maiores problemas dos DTD son:

- Non indica cal é o elemento raíz do documento (debe figurar na etiqueta DOCTYPE).
- Ausencia de tipos de datos: non é posible dicirlle, por exemplo:
  - ♦ que o contido da etiqueta <cantidad> sexa un número positivo.
  - ♦ que o contido da etiqueta <mes> sexa un número enteiro entre 1 e 12.
  - ♦ que o texto da etiqueta <title> sexa unha cadea de ata 255 caracteres.
- Non é unha linguaxe XML. Isto conleva a necesidade de ter distintos analizadores e API para o DTD e para o documento.
- Son moi pouco extensibles, e é difícil combinar diferentes DTD entre si.
- Non son apenas compatibles cos espazos de nomes XML.
- Cousas aparentemente sinxelas resólvense de xeito complicado: por exemplo, un elemento que contén a súa vez tres elementos pero non nos importa a orde.

O XML Schema Definition Language é unha recomendación do W3C que mellora en moitos aspectos a validación de documentos utilizando DTD.

Un **Schema XML** describe a estrutura dun documento XML. O Schema XML é unha alternativa ó uso do DTD. **NOTA:** Moitas veces a linguaxe empregada no Schema XML denomínase **XSD (XML Schema Definition)**.

### Exemplo de Schema XML.

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="correo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="destinatario" type="xs:string"/>
        <xs:element name="remitente" type="xs:string"/>
        <xs:element name="cabeceira" type="xs:string"/>
        <xs:element name="corpo" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

## ¿Que é un Schema XML?

O obxectivo dun Schema XML é definir a estrutura base dun documento XML, tal e como o fai un DTD.

Un Schema XML:

- define os elementos e atributos que poden aparecer nun documento
- define que elementos son elementos fillo
- define a orden dos elementos fillo
- define o número de elementos fillo
- define cando un elemento está baleiro ou cando pode incluír texto
- define tipos de datos para elementos e atributos
- define valores por defecto ou fixos para elementos e atributos

## Os Schemas XML son os sucesores dos DTD

As razóns polas que os Schemas están reemplazando os DTD son:

- Os Schemas XML pódense ampliar con futuras adicións
- Os Schemas XML son máis potentes e teñen máis riqueza cós DTD
- Os Schemas XML están escritos en XML
- Os Schemas XML soportan tipos de datos
- Os Schemas XML soportan espazos de nomes

En Maio do ano 2001 o Schema XML pasou a ser unha recomendación da organización W3C.

## Os Schemas XML soportan tipos de datos

Unha das grandes vantaxes dos Schemas XML é o soporte de tipos de datos. Grazas a isto é moito máis sinxelo:

- describir o contido permitido dentro dun documento
- validar a exactitude do documento
- traballar con datos dunha base de datos
- definir restricións de datos
- definir patróns de datos
- converter datos entre diferentes tipos

## Os Schemas XML usan sintaxe XML

Polo tanto:

- non temos que aprender unha nova linguaxe
- podemos empregar o editor XML para editar os nosos arquivos de schema
- podemos manipular o Schema có XML DOM
- podemos transformar o Schema con XSLT

## Os Schemas XML garanten a comunicación segura de datos

Cando enviamos datos dende un emisor a un receptor, é moi importante que ambas partes saiban como van a recibir e enviar eses datos.

Empregando os Schemas XML, o emisor pode describir os datos dun xeito que o receptor comprenda perfectamente. Por exemplo. Unha data tal como "09-06-1972", nalgúns países poderá ser interpretada como o 9 de xuño e noutros países o 6 de setembro.

Con todo, un elemento XML cun tipo de datos "date" como por exemplo:

```
<date type="date">1972-06-09</date>
```

asegura o entendemento mutuo entre o emisor e o receptor, xa que o tipo "date" en XML require o formato "YYYY-MM-DD".

## Ben formado non é suficiente

Un documento XML ben formado é un documento que cumpre as regras XML, como:

- comezar cunha declaración XML
- ter un único elemento raíz
- as marcas de comezo deben ter a correspondente marca de fin
- os elementos son sensibles a maiúsculas e minúsculas
- todos os elementos teñen que estar pechados
- todos os elementos deben estar correctamente aniñados
- todos os valores dos atributos teñen que estar entre comiñas
- para caracteres especiais deben usarse entidades

Incluso se os documentos están ben formados poden conter erros, e eses erros poden ter serias consecuencias.