

Tema 3

XML Schemas (1)

Contidos

Artigos

XML Schema	1
XML:Estrutura e enlace co XSD	1
XML:Compoñentes principais dun Schema	3
XML:Tipos simples predefinidos	4
XML:Declaraci•n de elementos	4
XML:Axustando a declaraci•n dun elemento	6

Referencias

Fonte dos artigos e contribu,ntes	7
-----------------------------------	---

Licenzas de artigos

Licenza	8
---------	---

XML Schema

O XML Schema Definition Language (XSDL ou XSD) , unha linguaxe XML (recomendado polo W3C) creada para describir a estrutura e restricións do contido dun documento XML.

Do mesmo xeito que os DTDs, serven para validar documentos XML, pero mellorando todos os problemas que estes tían.

O feito de que a súa linguaxe sexa tamén XML fai que poidamos utilizar as mesmas ferramentas que utilizamos para editar, representar ou transformar o documento orixinal.

XML:Estrutura e enlace co XSD

Estrutura e enlace co XSD

Veremos a continuación diversas posibilidades para crear o documento que contén o esquema, e o seu enlace cos documentos.

O feito de ser ambos documentos XML, fai que se realice unha dobre validación, posto que o documento .xml valídase no esquema.xsd, pero este a súa vez valídase nun hipotético XMLSchema.xsd, posto que o esquema tamén ten unha sintaxe definida no seu propio esquema-de-esquemas, e que ten que cumprir.

O documento XML non ten o seu propio espazo de nomes

Neste caso, simplemente indicamos no esquema un único espazo de nomes: o dos seus propios elementos. Isto débese a que os elementos e tipos de datos que usaremos para construír o esquema están definidos no espazo de nomes <http://www.w3.org/2001/XMLSchema> (ao que nos referiremos co prefixo xs:).

ESQUEMA.XSD

```
<?xml version="1.0" ... ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
    ... definición do esquema ...
</xs:schema>
```

Como o documento non ten o seu propio espazo de nomes, enlazarase ao seu esquema a través do atributo *xsi:noNamespaceSchemaLocation*. Para ter acceso a este atributo, creamos anteriormente o prefixo xsi: (*xml schema instance*) que se refire ao espazo de nomes especificado na etiqueta raíz.

DOCUMENTO.XML

```
<?xml version="1.0" ... ?>
<elementoRaíz xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="esquema.xsd">
    ...
</elementoRaíz>
```

Por exemplo: <toggledisplay>

ESQUEMACURSOS.XSD

```
<?xml version="1.0" ... ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" >
    <xs:element name="cursosEmpresa" type="xs:string"/> <!--
```

```

a definición de elementos verase posteriormente
...
</xs: schema>

```

CURSOS.XML

```

<?xml version="1.0" ... ?>
<cursosEmpresa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="EsquemaCursos.xsd">
  <nomeEmpresa>
    ...
  </cursosEmpresa>
</toggledisplay>

```

O documento XML ten o seu propio espazo de nomes

Se queremos que os elementos definidos neste esquema pertencen a un certo espazo de nomes, que usaremos nos documentos validados por devandito esquema, entón usaremos o atributo *targetNamespace* na definición do elemento raíz.

ESQUEMA.XSD

```

<?xml version="1.0" ... ?>
<xs: schema
  </nowiki>xmlns:xs="http://www.w3.org/2001/XMLSchema"
  targetNamespace="URI do EspazoDeNomes">
  ... definición do esquema ...
</xs: schema>

```

E no documento crearemos e utilizaremos nos elementos un prefixo para o espazo de nomes.

DOCUMENTO.XML

```

<?xml version="1.0" ... ?>
<pref: elementoRaíz
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:pref="URI do EspazoDeNomes" xsi:schemaLocation="URI do EspazoDeNomes
  esquema.xsd">
  ...
</pref: elementoRaíz>

```

Por exemplo: <toggledisplay>

ESQUEMACURSOS.XSD

```

<?xml version="1.0"?>
<xs: schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  targetNamespace="http://www.xunta.es/cursos">
    <xs: element name="cursosEmpresa" type="xs:string">
      <!-- a definición de elementos verase posteriormente -->
    ...
  </xs: schema>

```

CURSOS.XML

```
<?xml version="1.0"?>
<xg: cursosEmpresa
xml ns: xsi="http://www.w3.org/2001/XMLSchema-instance"
xml ns: xg="http://www.xunta.es/cursos"
xsi:schemaLocation="http://www.xunta.es/cursos/EsquemaCursos.xsd">
    <xg: nomeEmpresa> ...
    ...
</xg: cursosEmpresa>

</toggleDisplay>
```

XML:Componentes principais dun Schema

Componentes principais dun esquema

Existen catro construcións básicas nas que se basea un esquema:

- † Declaración de elementos .- Serve para asociar o nome dun elemento a un tipo predefinido, a un tipo simple ou a un tipo complexo.
- † Declaración de atributos .- Serve para asociar o nome dun atributo a un tipo simple (o valor dos atributos só pode conter texto).
- † Tipos simples .- Permitennos definir novos tipos baseados nos tipos simples predefinidos, aplicándolles certas restricións. O seu valor sempre , de tipo texto e non poden conter elementos (marcas <>), nin atributos.
- † Tipos complexos .- Serven para definir novos tipos de datos cuxos valores constan de atributos e/ou elementos (tamén poden incluír texto no seu contido).

XML: Tipos simples predefinidos

Tipos simples predefinidos

A continuación podemos ver algunos de los tipos incorporados (primitivos e derivados) más utilizados.

Categoría	Tipos
texto	xs:string, xs:anyURI, xs:normalizedString
números	xs:byte, xs:decimal, xs:double, xs:float, xs:int, xs:integer, xs:long, xs:negativeInteger (<0), xs:nonNegativeInteger (>=0), xs:positiveInteger (>0), xs:nonPositiveInteger (<=0), xs:short, xs:unsignedByte, xs:unsignedInt, xs:unsignedLong, xs:unsignedShort
data/hora	xs:date, xs:dateTime, xs:duration, xs:gDay, xs:gMonth, xs:gMonthDay, xs:gYear, xs:gYearMonth, xs:time
lógicos	xs:boolean

Cuando especifiquemos un contenido tendremos en cuenta que deberá seguir el formato aceptado por el tipo de dato elegido. Por ejemplo, con un número real utilizaremos el punto como separador decimal, con una fecha utilizaremos el formato "AAAA-MM-DD", y con una hora utilizaremos el formato "hh:mm:ss". El tipo boolean permite introducir *true* o 1, e *false* o 0, indistintamente.

XML: Declaración de elementos

Declaración de elementos

La declaración de un elemento XML utiliza, en la definición del esquema, la etiqueta `<xs:element ...>`. Dependiendo de la utilización del elemento en el documento, puede declararse de alguna de las siguientes formas.

Basados en un tipo predefinido

Los elementos basados en un tipo simple predefinido sólo pueden contener texto que cumpla las restricciones del tipo al que se basa. No se permiten ni subelementos ni atributos.

```
<xs:element name="nombreElemento" type="tipoPredefinido" />
```

Ejemplo: `<toggledisplay>`

A siguiente definición.

```
<xs:element name="nombreCurso" type="xs:string"/>
<xs:element name="numeroAsistentes" type="xs:nonNegativeInteger" />
```

Permitir la inclusión en un documento de los elementos:

```
<nombreCurso>Programación en C#</nombreCurso>
<numeroAsistentes>15</numeroAsistentes>

</toggledisplay>
```

Baseados nun Tipo Simple definido polo usuario

Os elementos baseados nun tipo simple só poden conter texto que cumpra as restricións impostas polo tipo simple definido polo usuario. Non se permiten nin (sub)elementos nin atributos.

É igual que no caso dos tipos predefinidos (que se consideran tipos simples), pero especificando a maiores as condicións que deben cumprir os datos.

A definición do tipo simple pódese realizar no interior da declaración do elemento, polo que quedará oculta ao resto do esquema, ou fora, para poder utilizarse en varias declaracións.

```
<xs:simpleType name="nomeDoTipoSimple">
    ...
</xs:simpleType>
<xs:element name="nomeElemento" type="nomeDoTipoSimple"/>
```

ou ben:

```
<xs:element name="nomeElemento">
    <xs:simpleType>
        ...
    </xs:simpleType>
</xs:element>
```

Baseados nun Tipo Complexo definido polo usuario

Cando un elemento conteña atributos e/ou sub-elementos no seu contido, entón será necesario crear un tipo complexo.

A declaración dun elemento baseado nun tipo complexo, exactamente igual a dun tipo simple, trocando `<xs:simpleType>` por `<xs:complexType>`. A diferenza, evidentemente, a propia declaración destes tipos, que veremos en posteriores epígrafes.

XML:Axustando a declaración dun elemento

Axustando a declaración dun elemento

Na declaración dun elemento `<xs:element ... >` podemos especificar diversos atributos que nos permitirán configuralo de xeito adecuado:

- † `name`: o nome do elemento, o que se usará nos documentos XML encerrado entre os símbolos `<>`.
- † `type`: o nome do tipo de datos do contido do elemento. Como xa se comentou pode ser un tipo predefinido, un tipo simple ou un tipo complexo.
- † `maxOccurs`: número máximo de veces que o elemento pode aparecer dentro do elemento que o contén. O seu valor debe ser de tipo *nonNegativeInteger* ou ben a cadea **unbounded** para que non teña límite. Por defecto, 1.
- † `minOccurs`: número mínimo de veces que o elemento debe aparecer dentro do elemento que o contén. O seu valor debe ser de tipo *nonNegativeInteger*. Se especificamos o valor 0 convértese nun elemento opcional. Por defecto, 1.
- † `default`: o valor predeterminado para o contido do elemento, se este non se especifica no documento (non compatible con *fixed*, a continuación).
- † `fixed`: o seu valor, automaticamente asignado ao contido do elemento, sen poder cambiarse no documento (non compatible con *default*).
- † `ref`: o seu valor, o nome doutro elemento definido no esquema (ou noutro esquema especificando o seu espazo de nomes). Se especificamos este atributo, a definición do elemento non pode conter ademais os atributos *default*, *fixed* ou *type* (entre outros), nin conter os elementos `<simpleType>` ou `<complexType>`.

Exemplos do uso destes atributos poderían ser os seguintes: `<toggle display>`

```
<xs:element name="dataDeNacemento" type="xs:date" minOccurs="0"/>
<xs:element name="lixa" type="xs:string" maxOccurs="unbounded"/>
<xs:element name="numeroDeCopias" type="xs:positiveInteger"
default="1"/>
<xs:element ref="nomeDoElementoReferenciado" />
```

`</toggle display>`

Fonte dos artigos e contribuintes

XML Schema `Fonte: <http://www.plategaxml.es/index.php?oldid=508> `Contribuintes: Victor

XML: Estrutura e enlace co XSD `Fonte: <http://www.plategaxml.es/index.php?oldid=615> `Contribuintes: Victor

XML: Componentes principais dun Schema `Fonte: <http://www.plategaxml.es/index.php?oldid=680> `Contribuintes: David, Victor

XML: Tipos simples predefinidos `Fonte: <http://www.plategaxml.es/index.php?oldid=801> `Contribuintes: David, Victor

XML: Declaración de elementos `Fonte: <http://www.plategaxml.es/index.php?oldid=678> `Contribuintes: David, Victor

XML: Axustando a declaración dun elemento `Fonte: <http://www.plategaxml.es/index.php?oldid=821> `Contribuintes: Victor

Licenza

Attribution-Noncommercial-Share Alike 3.0 Unported
<http://creativecommons.org/licenses/by-nc-sa/3.0/>