

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:Declaración de elementos	4
XML:Axustando a declaración dun elemento	6

Referencias

Fonte dos artigos e contribuíntes	7
-----------------------------------	---

Licenzas de artigos

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

XML Schema

O XML Schema Definition Language (XSDL ou XSD) é 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 tiñan.

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 dúas posibilidades para crear o documento que contén o esquema, e o seu enlace cos documentos.

O feito de ser ámbos documentos XML, fai que se realice unha dobre validación, posto que o documento .xml validase no esquema.xsd, pero este á súa vez validase 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, enlazárase 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 pertencan 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="URIdolEspazoDeNomes">
  ... 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="URIdoEspazoDeNomes" xsi:schemaLocation="URIdoEspazoDeNomes
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
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xg="http://www.xunta.es/cursos"
xsi:schemaLocation="http://www.xunta.es/cursos/EsquemaCursos.xsd">
    <xg:nomeEmpresa> ...
    ...
</xg:cursosEmpresa>
```

</toggedisplay>

XML:Compoñentes principais dun Schema

Compoñentes 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 .- Permítennos definir novos tipos baseados nos tipos simples predefinidos, aplicándolles certas restricións. O seu valor sempre é 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 algúns dos tipos incorporados (primitivos e derivados) máis 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

Cando especifiquemos un contido teremos en conta que terá que seguir o formato axeitado ó tipo de dato elixido. Por exemplo, cun número real utilizaremos o punto como separador decimal, cunha data utilizaremos o formato "AAAA-MM-DD", e cunha hora utilizaremos o formato "hh:mm:ss". O tipo boolean permite introducir *true* ou 1, e *false* ou 0, indistintamente.

XML:Declaración de elementos

Declaración de elementos

A declaración dun elemento XML utiliza, na definición do esquema, a etiqueta `<xs:element ...>`. Dependendo da utilización do elemento no documento, pode declararse dalgún dos seguintes xeitos.

Baseados nun tipo predefinido

Os elementos baseados nun tipo simple predefinido só poden conter texto que cumpra as restricións do tipo no que se basea. Non se permiten nin (sub)elementos nin atributos.

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

Exemplo: `<toggledisplay>`

A seguinte definición.

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

Permitiría incluír nun documento os elementos:

```
<nomeCurso>Programación en C#</nomeCurso>
<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 quedaría 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 é exactamente igual á dun tipo simple, trocando `<xs:simpleType>` por `<xs:complexType>`. A diferenza é, 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 á súa vez o contén. O seu valor debe ser de tipo *nonNegativeInteger* ou ben a cadea **unbounded** para que non teña límite. Por defecto é 1.
- **minOccurs**: número mínimo de veces que o elemento debe aparecer dentro do elemento que á súa vez o contén. O seu valor debe ser de tipo *nonNegativeInteger*. Se especificamos o valor 0 convértese nun elemento opcional. Por defecto é 1.
- **default**: é 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 é automaticamente asignado ao contido do elemento, sen poder cambiarse no documento (non compatible con *default*).
- **ref**: o seu valor é 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="liña" type="xs:string" maxOccurs="unbounded"/>
<xs:element name="númeroDeCopias" type="xs:positiveInteger"
default="1"/>
<xs:element ref="nomeDoElementoReferenciado" />
```

`</toggle display>`

Fonte dos artigos e contribuíntes

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

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

XML:Compoñentes principais dun Schema *Fonte:* <http://www.plategaxml.es/index.php?oldid=680> *Contribuíntes:* David, Victor

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

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

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

Licenza

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