

# Tipos simples

## Sumario

- 1 Elementos XSD
  - ◆ 1.1 Definindo un elemento simple
  - ◆ 1.2 Valores fixos e por defecto para elementos simples
- 2 Atributos XSD
  - ◆ 2.1 Que é un atributo?
  - ◆ 2.2 Como definir un atributo
  - ◆ 2.3 Valores fixos e por defecto nos atributos
  - ◆ 2.4 Atributos opcionais e requiridos
  - ◆ 2.5 Restricións nos atributos
- 3 Restricións XSD
  - ◆ 3.1 Restricións en valores
  - ◆ 3.2 Restricións nun conxunto de valores
  - ◆ 3.3 Restricións nunha serie de valores
  - ◆ 3.4 Outras restricións nunha serie de valores
  - ◆ 3.5 Restricións de lonxitude
  - ◆ 3.6 Restricións para tipos de datos

## Elementos XSD

Os Schemas XML definen os elementos dos arquivos XML.

Un **elementos simple** é un elemento XML que contén soamente texto. Non pode conter calquera outros elementos ou atributos.

### Definindo un elemento simple

A sintaxe para definir un elemento simple nun XSD é:

```
<xs:element name="xxx" type="yyy"/>
```

Onde xxx é o nome do elemento e yyy é o tipo de datos do elemento.

Os Schemas XML teñen unha morea de tipos de datos. Os máis comúns son:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

### Valores fixos e por defecto para elementos simples

Os elementos simples poden ter un valor por defecto ou un valor fixo.

O **valor por defecto** é asignado automaticamente ó elemento cando non se especifica outro valor. No seguinte código de exemplo o valor por defecto é vermello:

```
<xs:element name="color" type="xs:string" default="vermello"/>
```

Un **valor fixo** é asignado automaticamente ó elemento e non se pode especificar outro valor diferente. No seguinte exemplo o valor fixo é "vermello":

```
<xs:element name="color" type="xs:string" fixed="vermello"/>
```

## Atributos XSD

Todos os atributos son declarados como tipos simples.

### Que é un atributo?

Os elementos simples non poden ter atributos. Si un elemento ten atributos entón será considerado un elemento complexo. Pero o atributo en si mesmo é sempre declarado como un tipo simple.

### Como definir un atributo

A sintaxe para definir un atributo é:

```
<xs:attribute name="xxx" type="yyy"/>
```

onde xxx é o nome do atributo e yyy especifica o tipo de datos do atributo.

Os Schemas XML teñen un montón de tipos de datos predefinidos. Os máis comúns son:

- xs:string
- xs:decimal
- xs:integer
- xs:boolean
- xs:date
- xs:time

Exemplo:

```
<apellido idioma="ES">Losada</apellido>
```

E aquí está a correspondente definición do atributo:

```
<xs:attribute name="idioma" type="xs:string"/>
```

### Valores fixos e por defecto nos atributos

Os atributos poden ter un valor por defecto ou un valor fixo.

O valor por defecto será asignado cando non se especifique ningún outro valor.

No seguinte exemplo o valor por defecto é "ES":

```
<xs:attribute name="idioma" type="xs:string" default="ES"/>
```

Para asignar un valor fixo empregamos a palabra fixed, e non poderemos especificar outro valor. No seguinte exemplo o valor fixo é "ES":

```
<xs:attribute name="idioma" type="xs:string" fixed="ES"/>
```

### Atributos opcionais e requiridos

Os atributos son opcionais por defecto. Para especificar cun atributo é requirido ou obrigatorio, temos que empregar o atributo "use":

```
<xs:attribute name="idioma" type="xs:string" use="required"/>
```

### Restricións nos atributos

Cando un elemento XML ou atributo ten un tipo de datos definido, está engadindo restricións no elemento ou no contido do atributo.

Por exemplo si un elemento ten un tipo de datos "xs:date" e contén unha cadea como "Ola mundo", o elemento non validará.

Si desexamos engadir restricións dun determinado tipo a un atributo, como por exemplo que o valor sexa un número decimal maior que 1 poderémolo facer empregando restricións.

Neste exemplo temos un atributo de nome `cantidad` que pode almacenar números decimais, no cal lle indicamos que o valor máis pequeno que se pode introducir será 1.

```
<xs:attribute name="cantidad">
  <xs:simpleType>
    <xs:restriction base="xs:decimal">
      <xs:minInclusive value="1"/>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
```

Máis información das restricións nos Schemas XSD a continuación.

## Restricións XSD

As restricións son usadas para definir os valores aceptables para os elementos XML ou os atributos.

### Restricións en valores

O seguinte exemplo define un elemento chamado `idade` cunha restrición. O valor da idade non pode ser menos que 0 ou maior que 120:

```
<xs:element name="idade">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="0"/>
      <xs:maxInclusive value="120"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

### Restricións nun conxunto de valores

Para limitar o contido dun elemento XML a un conxunto aceptable de valores, empregaremos a restrición **enumeration**.

Por exemplo, neste código definimos un elemento chamado `coche` cunha restrición. Os únicos valores aceptables son: BMW, Audi, Golf:

```
<xs:element name="coche">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="BMW"/>
      <xs:enumeration value="Audi"/>
      <xs:enumeration value="Golf"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O exemplo anterior tamén se podería escribir así:

```
<xs:element name="coche" type="TipoCoche"/>

<xs:simpleType name="TipoCoche">
  <xs:restriction base="xs:string">
    <xs:enumeration value="BMW"/>
    <xs:enumeration value="Audi"/>
    <xs:enumeration value="Golf"/>
  </xs:restriction>
</xs:simpleType>
```

**NOTA:** neste caso o tipo de datos `"TipoCoche"` poderá ser empregado por outros elementos xa que non forma parte do elemento `"coche"`.

## Restricións nunha serie de valores

Para limitar o contido dun elemento XML a unha serie de números ou letras, empregaremos a restrición **pattern**.

O seguinte exemplo define o elemento letra cunha restrición. Os únicos valores aceptables son UNHA das letras do abecedario entre a-z (en minúsculas):

```
<xs:element name="letra">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define unha elemento chamado "iniciais" cunha restrición. O único valor aceptado son 3 letras en maiúsculas dende a ata z:

```
<xs:element name="iniciais">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[A-Z][A-Z][A-Z]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define de novo outro elemento chamado "iniciais" cunha restrición. Soamente se aceptan 3 letras en maiúsculas ou minúsculas dende a ata z:

```
<xs:element name="iniciais">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z][a-zA-Z][a-zA-Z]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define un elemento chamado "opcion" cunha restrición. Soamente se acepta UNHA das seguintes letras: x, y ou z:

```
<xs:element name="opcion">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[xyz]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define un elemento chamado identificador cunha restrición. Soamente se aceptan 5 díxitos en secuencia, e cada dígito estará no rango de 0 ata 9:

```
<xs:element name="identificador">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:pattern value="[0-9][0-9][0-9][0-9][0-9]" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Outras restricións nunha serie de valores

O seguinte exemplo define un elemento chamado "letras" cunha restrición. O valor aceptado pode ser 0 ou máis ocorrencias das letras do abecedario en minúsculas:

```
<xs:element name="letras">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-z]*" />
    </xs:restriction>
  </xs:simpleType>
```

```
</xs:element>
```

O seguinte exemplo define un elemento chamado sexo cunha restrición. Os únicos valores aceptados son home ou muller:

```
<xs:element name="sexo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="home|muller"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define un elemento chamado "contrasinal" cunha restrición. Debe ter exactamente 8 caracteres, en minúsculas ou maiúsculas con letras de a-z ou números 0-9:

```
<xs:element name="contrasinal">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern value="[a-zA-Z0-9]{8}"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Restricións de lonxitude

Para limitar a lonxitude dun valor nun elemento, podemos empregar as restricións maxLength, minLength e length.

O seguinte exemplo amosa un elemento chamado "contrasinal" cunha restrición. O valor debe ter exactamente 8 caracteres:

```
<xs:element name="contrasinal">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

O seguinte exemplo define outro elemento "contrasinal" cunha restrición. O valor debe ter como mínimo 5 caracteres e como máximo 8 caracteres:

```
<xs:element name="contrasinal">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:minLength value="5"/>
      <xs:maxLength value="8"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

## Restricións para tipos de datos

Restrición	Descrición
enumeration	Define os valores permitidos nunha lista
fractionDigits	Especifica o número máximo de decimais permitidos. Debe ser maior ou igual a 0.
length	Especifica o número exacto de caracteres ou elementos permitidos. Debe ser maior ou igual a 0.
maxExclusive	Especifica o límite superior para valores numéricos (o valor debe ser menor que este límite)
maxInclusive	Especifica o límite superior para valores numéricos (o valor debe ser menor ou igual que este límite)
maxLength	Especifica o número máximo de caracteres ou valores permitidos. Debe ser maior ou igual que cero.
minExclusive	Especifica o límite inferior para valores numéricos (o valor debe ser maior que este límite)

Restrición	Descrición
minInclusive	Especifica o límite inferior para valores numéricos (o valor debe ser menor ou igual a este límite)
minLength	Especifica o número mínimo de caracteres ou valores permitidos. Debe ser maior ou igual que cero.
pattern	Define a secuencia exacta de caracteres permitidos.
totalDigits	Define o número exacto de díxitos permitidos. Debe ser maior que cero.
whiteSpace	Especifica como se xestionan os espazos en branco (tabs, saltos de liña, etc.)