

# Tipos de datos

## Sumario

- 1 Cadeas XSD
  - ◆ 1.1 Tipo datos NormalizedString
  - ◆ 1.2 Tipo datos Token
  - ◆ 1.3 Tipos de datos en String
  - ◆ 1.4 Restricións en tipos de datos String
- 2 Datos XSD
  - ◆ 2.1 Tipo Date
  - ◆ 2.2 Tipo Time
  - ◆ 2.3 Tipo DateTime
  - ◆ 2.4 Tipo Duracion
  - ◆ 2.5 Tipos de datos en Date e Time
  - ◆ 2.6 Restricións nos tipos de datos Date
- 3 Números XSD
  - ◆ 3.1 Tipo Decimal
  - ◆ 3.2 Tipo Integer
  - ◆ 3.3 Tipos de datos numéricos
  - ◆ 3.4 Restricións en tipos de datos numéricos
- 4 Miscelánea
  - ◆ 4.1 Tipo Boolean
  - ◆ 4.2 Tipo Binary
  - ◆ 4.3 Tipo AnyURI
  - ◆ 4.4 Restricións en tipos de datos miscelaneos

## Cadeas XSD

Os tipos de datos cadea (**string**) poden conter caracteres, saltos de liña, tabulacións, etc.

O seguinte é un exemplo de declaración cadea:

```
<xs:element name="cliente" type="xs:string"/>
```

Un elemento no documento XML pode ser como:

```
<cliente>John Smith</cliente>
```

Ou podería ser como:

```
<cliente>      John Smith      </cliente>
```

**NOTA:** O procesador XML non modificará o valor se empregas o tipo string.

### Tipo datos NormalizedString

Este tipo é un derivado do tipo de datos String. O tipo de datos **normalizedString** tamén contén caracteres, pero o procesador XML eliminará os saltos de liña, comezos de liña, caracteres de tabulación, etc.

O seguinte é un exemplo de declaración cadea:

```
<xs:element name="cliente" type="xs:normalizedString"/>
```

Un elemento no documento XML pode ser como:

```
<cliente>John Smith</cliente>
```

Ou podería ser como:

```
<cliente>      John Smith      </cliente>
```

**NOTA:** No exemplo anterior o procesador XML reemplazará os tabuladores con espazos.

**Tipo datos Token**

O tipo de datos token tamén está derivado do tipo de datos String.

O tipo de datos token contén tamén caracteres, pero o procesador XML eliminará os saltos de liña, tabuladores, espazos múltiples,etc.

Exemplo de declaración token nun esquema:

```
<xs:element name="cliente" type="xs:token"/>
```

Un elemento no documento XML pode ser como:

```
<cliente>John Smith</cliente>
```

Ou podería ser como:

```
<cliente>      John Smith      </cliente>
```

**NOTA:** No exemplo anterior o procesador XML eliminaría os tabuladores.

**Tipos de datos en String**

Nome	Descrición
ENTITIES	
ENTITY	Define un valor de data e hora
ID	Unha cadea que representa o atributo ID en XML
IDREF	Unha cadea que representa o atributo IDREF en XML
IDREFS	
language	Unha cadea que contén un id de linguaxe válido.
Name	Unha cadea que contén un nome XML válido.
NCName	
NMTOKEN	Unha cadea que representa o atributo NMTOKEN en XML (usado soamente con atributos de schema).
NMTOKENS	
normalizedString	Unha cadea que non contén saltos de liña, saltos de carro ou tabuladores.
QName	
string	Unha cadea
token	Unha cadea que non contén saltos de liña, de carro, tabuladores, espazos iniciais e finais ou múltiples espazos.

Cando queremos incluír texto especial como outras entidades por exemplo, e queremos que ese texto non sexa interpretado como outro elemento do XML, empregaremos **<![CDATA[<]]>**.

Por exemplo:

```
<![CDATA[Se desenvolves en HTML5 xa podes empregar as novas etiquetas como:  
<video> </video>, <audio> </audio>, <canvas> </canvas>, etc.]]>
```

## Restricións en tipos de datos String

As restricións que poden ser usadas cós tipos de datos String:

- enumeration
- length
- maxLength
- minLength
- pattern (NMTOKENS, IDREFS, e ENTITIES non poden usar esta restrición)
- whiteSpace

## Datas XSD

Os tipos de datos data e hora úsanse para especificar datas.

### Tipo Date

A data é especificada do xeito seguinte: "YYYY-MM-DD" onde:

- YYYY indica o ano
- MM indica o mes
- DD indica o día

O seguinte é un exemplo de declaración nun schema:

```
<xs:element name="comezo" type="xs:date"/>
```

O elemento no noso XML podería quedar como:

```
<comezo>2002-09-24</comezo>
```

**Time Zones:** Para especificar unha zoa horaria podemos introducir a data en formato UTC engadindo unha "Z" ó final da data:

```
<comezo>2002-09-24Z</comezo>
```

Ou se podería especificar un desplazamento de tempo engadindo unha hora positiva ou negativa, como por exemplo:

```
<comezo>2002-09-24-06:00</comezo>  
ou  
<comezo>2002-09-24+06:00</comezo>
```

### Tipo Time

A hora é especificada do xeito seguinte: "hh:mm:ss" onde:

- hh indica a hora
- mm indica os minutos
- ss indica os segundos

**Nota:** todos os compoñentes son obrigatorios.

O seguinte é un exemplo de declaración nun schema:

```
<xs:element name="comezo" type="xs:time"/>
```

O elemento no noso XML podería quedar como:

```
<comezo>09:00:00</comezo>
```

Ou podería ser como:

```
<comezo>09:30:10.5</comezo>
```

**Time Zones:** Para especificar unha zoa horaria podemos introducir a data en formato UTC engadindo unha "Z" ó final da data:

```
<comezo>09:00:00Z</comezo>
```

Ou se podería especificar un desplazamento de tempo engadindo unha hora positiva ou negativa, como por exemplo:

```
<comezo>09:30:10-06:00</comezo>  
ou  
<comezo>09:30:10+06:00</comezo>
```

## Tipo DateTime

O tipo de datos DateTime úsase para especificar datas e horas.

A data e hora é especificada do xeito seguinte: "YYYY-MM-DDThh:mm:ss" onde:

- YYYY indica o ano
- MM indica o mes
- DD indica o día
- T indica o comezo da sección de tempo
- hh indica a hora
- mm indica os minutos
- ss indica os segundos

Nota: todos os compoñentes son obrigatorios.

O seguinte é un exemplo de declaración nun schema:

```
<xs:element name="comezo" type="xs:dateTime"/>
```

O elemento no noso XML podería quedar como:

```
<comezo>2002-05-30T09:00:00</comezo>
```

Ou podería ser tamén:

```
<comezo>2002-05-30T09:30:10.5</comezo>
```

**Time Zones:** Para especificar unha zoa horaria podemos introducir a data en formato UTC engadindo unha "Z" ó final da data:

```
<comezo>2002-05-30T09:30:10Z</comezo>
```

Ou se podería especificar un desplazamento de tempo engadindo unha hora positiva ou negativa, como por exemplo:

```
<comezo>2002-05-30T09:30:10-06:00</comezo>  
ou  
<comezo>2002-05-30T09:30:10+06:00</comezo>
```

## Tipo Duracion

O tipo de datos duracion é empregado para indicar un intervalo.

O intervalo de tempo indícase por "PnYnMnDTnHnMnS" onde:

- P indica o período (obligatorio)
- nY indica o número de anos
- nM indica o número de meses

- nD indica o número de días
- T indica o comezo da sección de tempo (obligatorio si queremos especificar horas, minutos e segundos)
- nH indica o número de horas
- nM indica o número de minutos
- nS indica o número de segundos

O seguinte exemplo é unha declaración de período nun schema:

```
<xs:element name="periodo" type="xs:duration"/>
```

- Un elemento no documento podería ser como:

```
<periodo>P5Y</periodo>
```

Este exemplo indica un período de 5 anos.

- Ou podería ser como:

```
<periodo>P5Y2M10D</periodo>
```

Indica un período de 5 anos, 2 meses e 10 días.

- Ou podería ser como:

```
<periodo>P5Y2M10DT15H</periodo>
```

Indica un período de 5 anos, 2 meses, 10 días e 15 horas.

- Ou podería ser como:

```
<periodo>PT15H</periodo>
```

Indica un período de 15 horas.

- Duración negativa: Para especificar unha duración negativa, temos que usar un signo - antes de P:

```
<periodo>-P10D</periodo>
```

Indica un período de menos 10 días.

## Tipos de datos en Date e Time

Nome	Descrición
date	Define un valor de data
dateTime	Define un valor de data e hora
duration	Define un intervalo de tempo
gDay	Define unha parte dunha data - o día (DD)
gMonth	Define unha parte dunha data - o mes (MM)
gMonthDay	Define unha parte dunha data - o mes e o día (MM-DD)
gYear	Define unha parte dunha data - o ano (YYYY)

Nome	Descrición
gYearMonth	Define unha parte dunha data - o ano e o mes (YYYY-MM)
time	Define un valor de tempo

## Restricións nos tipos de datos Date

As restricións que se poden usar con tipos de datos Date:

- enumeration
- maxExclusive
- maxInclusive
- minExclusive
- minInclusive
- pattern
- whiteSpace

## Números XSD

Vexamos como especificar os tipos de datos numéricos.

### Tipo Decimal

O tipo decimal emprégase para indicar un valor numérico.

O seguinte é un exemplo dunha declaración decimal nun schema:

```
<xs:element name="prezo" type="xs:decimal"/>
```

Un elemento XML podería ser como:

```
<prezo>999.50</prezo>
```

Ou podería ser como:

```
<prezo>+999.5450</prezo>
```

Ou como:

```
<prezo>-999.5230</prezo>
```

Ou como:

```
<prezo>0</prezo>
```

Ou como:

```
<prezo>0</prezo>
```

**Nota:** O número máximo de díxitos decimais permitidos é de 18.

## Tipo Integer

O tipo integer emprégase para indicar un valor numérico sin componentes fraccionais.

O seguinte é un exemplo dunha declaración integer nun schema:

```
<xs:element name="prezo" type="xs:integer"/>
```

Un elemento XML podería ser como:

```
<prezo>999</prezo>
```

Ou podería ser como:

```
<prezo>+999</prezo>
```

Ou como:

```
<prezo>-999</prezo>
```

Ou como:

```
<prezo>0</prezo>
```

## Tipos de datos numéricos

Todos os tipos de datos desta táboa derivan do tipo de dato Decimal.

Nome	Descrición
byte	Un enteiro con signo de 8-bit
decimal	Un valor decimal
int	Un enteiro con signo de 32-bit
integer	Un valor enteiro
long	Un enteiro con signo de 64-bit
negativeInteger	Un enteiro que contén soamente valores negativos (...,-2,-1)
nonNegativeInteger	Un enteiro que contén soamente valores non negativos (0,1,2,...)
nonPositiveInteger	Un enteiro que contén soamente valores non positivos (...,-2,-1,0)
positiveInteger	Un enteiro que contén soamente valores positivos (1,2,...)
short	Un enteiro de 16-bit
unsignedLong	Un enteiro sen signo de 64-bit
unsignedInt	Un enteiro sen signo de 32-bit
unsignedShort	Un enteiro sen signo de 16-bit
unsignedByte	Un enteiro sen signo de 8-bit

## Restricións en tipos de datos numéricos

As restricións que se poden usar cos tipos numéricos son:

- enumeration
- fractionDigits
- maxExclusive
- maxInclusive
- minExclusive
- minInclusive
- pattern
- totalDigits
- whiteSpace

## Miscelánea

Outros tipos de datos a maiores poden ser boolean, base64Binary, hexBinary, float, double, anyURI, QName e NOTATION.

### Tipo Boolean

Este tipo de datos emprégase para especificar un valor true ou false.

Exemplo de declaracion boolean nun schema:

```
<xs:attribute name="deshabilitado" type="xs:boolean"/>
```

Un elemento no documento xml podería ser:

```
<prezo deshabilitado="true">999</prezo>
```

**Nota:** tamén se pode empregar 0 (false) ou 1(true) como valores booleanos.

### Tipo Binary

Os tipos de datos binarios empréganse para formatear datos binarios.

Temos dous tipos de datos binarios:

- base64Binary (Base64-encoded binary data)
- hexBinary (hexadecimal-encoded binary data)

Exemplo de declaración de datos binarios nun schema:

```
<xs:element name="fotografia" type="xs:hexBinary"/>
```

### Tipo AnyURI

Este tipo de dato emprégase para especificar unha URL.

Exemplo de definición nun schema:

```
<xs:attribute name="src" type="xs:anyURI"/>
```

Exemplo de documento XML:

```
<pic src="http://www.iessanclemente.net/logo.jpg" />
```

**Nota:** Si a URL ten espazos hay que reemplazalos por %20



## Restricións en tipos de datos miscelaneos

Listado de restricións que se poden usar con estes tipos de datos:

- enumeration (un tipo de datos Boolean non pode usar esta restrición)
- length (un tipo de datos Boolean non pode usar esta restrición)
- maxLength (un tipo de datos Boolean non pode usar esta restrición)
- minLength (un tipo de datos Boolean non pode usar esta restrición)
- pattern
- whiteSpace