

**Tutorial de XSD
(XML Schema)**

- Guion del tutorial
- Qué es XSD
- Validación de un documento XML con XSD
- Elementos simples
- Atributos
- Restricciones (facetas)
- Extensiones
- Elementos complejos
- Indicadores

Material extra

- Apuntes de XSD
- Ejercicios resueltos
- Enunciados de los ejercicios resueltos

Otros tutoriales

- Tutorial de DTD
- Tutorial de XML
- {+}

[Inicio](#) » [Tutorial de XSD](#) » [Restricciones](#)

Restricciones (facetas) en XSD (XML Schema)

XML Schema permite definir restricciones a los posibles valores de los tipos de datos. Dichas restricciones se pueden establecer en diferentes aspectos, llamados facetas.

Dicho de otro modo, **las facetas permiten definir restricciones sobre los posibles valores de atributos o elementos**. Las facetas que pueden utilizarse son:

Facetas en XSD	
Faceta	Descripción
xs:length	Especifica una longitud fija.
xs:minLength	Especifica una longitud mínima.
xs:maxLength	Especifica una longitud máxima.
xs:pattern	Especifica un patrón de caracteres admitidos.
xs:enumeration	Especifica una lista de valores admitidos.
xs:whiteSpace	Especifica cómo se debe tratar a los posibles espacios en blanco, las tabulaciones, los saltos de línea y los retornos de carro que puedan aparecer.
xs:maxInclusive	Especifica que el valor debe ser menor o igual que el indicado.
xs:maxExclusive	Especifica que el valor debe ser menor que el indicado.
xs:minExclusive	Especifica que el valor debe ser mayor que el indicado.
xs:minInclusive	Especifica que el valor debe ser mayor o igual que el indicado.
xs:totalDigits	Especifica el número máximo de dígitos que puede tener un número.
xs:fractionDigits	Especifica el número máximo de decimales que puede tener un número.

Seguidamente, se muestran algunos ejemplos de restricciones definidas con una o más facetas:

xs:minExclusive y xs:maxInclusive

EJEMPLO En el siguiente código se define un elemento llamado "mes" con la restricción de que el valor que tome no pueda ser menor que 1 ni mayor que 12:

```
<xs:element name="mes">
  <xs:simpleType>
    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1"/>
      <xs:maxInclusive value="12"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **xs:simpleType** permite definir un tipo simple y especificar sus restricciones.
- **xs:restriction** sirve para definir restricciones de un **xs:simpleType** (como se ha hecho en este ejemplo). También sirve para definir restricciones de un **xs:simpleContent** o de un **xs:complexContent**. Estos elementos se estudiarán más adelante en este tutorial.
- En el atributo **base** se indica el tipo de dato a partir del cual se define la restricción.
- **xs:minInclusive** sirve para especificar que el valor debe ser mayor o igual que el indicado en su atributo **value**, (en este caso, mayor o igual que 1).
- **xs:maxInclusive** sirve para especificar que el valor debe ser menor o igual que el indicado en su atributo **value**, (en este caso, menor o igual que 12).

También se podría haber escrito:

```
<xs:element name="mes" type="numeroMes"/>

<xs:simpleType name="numeroMes">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="1"/>
    <xs:maxInclusive value="12"/>
  </xs:restriction>
</xs:simpleType>
```

Haciendo esto, el tipo **numeroMes** definido, podría ser utilizado por otros elementos, ya que, no está contenido en el elemento "mes".

Ejercicios resueltos

- Edad entre 0 y 130 años
- Precios de tres dígitos

xs:enumeration

EJEMPLO En el siguiente ejemplo se define un elemento llamado "color" con la restricción de que los únicos valores admitidos son: **"verde"**, **"amarillo"** y **"rojo"**.

```
<xs:element name="color">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="verde"/>
      <xs:enumeration value="amarillo"/>
      <xs:enumeration value="rojo"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **xs:enumeration** sirve para definir una lista de valores admitidos.

Ejercicio resuelto

- Tipo de vehículo

xs:pattern

EJEMPLO En el siguiente ejemplo se define un elemento llamado "letra" con la restricción de que el único valor admitido es una de las letras minúsculas de la "a" a la "z":

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

- **xs:pattern** sirve para definir un patrón de caracteres admitidos (en este caso se admite una única letra minúscula de la "a" a la "z"). **El valor del patrón tiene que ser una expresión regular.**

Ejercicios resueltos

- Iniciales de personas famosas
- Iniciales al revés
- Respuestas admitidas
- Números y letras
- Escribir expresiones regulares
- Letras admitidas

xs:length

EJEMPLO En el siguiente ejemplo se define un elemento llamado "clave" con la restricción de que su valor tiene que ser una cadena de, exactamente, doce caracteres:

```
<xs:element name="clave">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:length value="12"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **xs:length** sirve para especificar una longitud fija.

Ejercicios resueltos

- Longitud fija de una clave
- Longitud mínima y máxima de una clave

xs:whiteSpace

EJEMPLO En el siguiente ejemplo se define un elemento llamado "dirección" con la restricción de que los espacios en blanco, las tabulaciones, los saltos de línea y los retornos de carro que aparezcan en él, se deben mantener (**preserve**):

```
<xs:element name="direccion">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:whiteSpace value="preserve"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

- **xs:whiteSpace** sirve para especificar cómo se debe tratar a los posibles espacios en blanco, las tabulaciones, los saltos de línea y los retornos de carro que puedan aparecer.

En vez de **preserve** también se puede utilizar:

- **replace** para sustituir todas las tabulaciones, los saltos de línea y los retornos de carro por espacios en blanco.
- **collapse** para, después de reemplazar todas las tabulaciones, los saltos de línea y los retornos de carro por espacios en blanco, eliminar todos los espacios en blanco únicos y sustituir varios espacios en blanco seguidos por un único espacio en blanco.

[Anterior](#) «» [Siguiente](#)

[Inicio](#) [Guía de estudios](#) [Tutoriales](#) [Diccionario](#) [Recursos](#) [Secciones](#) [Ayudar](#) [Contactar](#) [Condiciones](#) [Privacidad](#) [Licencia CC](#) [Acerca de](#) [Mapa Web](#)

© Abrirllave.com