

Restricciones (facetas)

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. Además, añadir condiciones a alguno de los tipos predefinidos en el XML Schema es la forma más sencilla de crear un nuevo tipo a partir de uno ya existente. 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.

xs:minExclusive y xs:maxInclusive

"xs:minInclusive" sirve para especificar que el valor debe ser mayor o igual que el indicado en su atributo **"value"**.

"xs:maxInclusive" sirve para especificar que el valor debe ser menor o igual que el indicado en su atributo **"value"**.

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 el atributo “**base**” se indica el tipo de dato a partir del cual se define la restricción.

xs:enumeration

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

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:pattern

“**xs:pattern**” sirve para definir un patrón de caracteres admitidos, es decir, permite construir máscaras que han de cumplir los datos de un elemento. **El valor del patrón tiene que ser una expresión regular.** La siguiente tabla muestra algunos de los caracteres que tienen un significado especial para la generación de las máscaras.

Elementos para hacer patrones.

Patrón	Significado
[A-Z a-z]	Letra.
[A-Z]	Letra mayúscula.
[a-z]	Letra minúscula.
[0-9]	Dígitos decimales.
\D	Cualquier carácter excepto un dígito del 0 al 9.
(A)	Cadena que coincide con A.

A B	Cadena que es igual a la cadena A o a la B.
AB	Cadena que es la concatenación de las cadenas A y B.
A?	Cero o una vez la cadena A.
A+	Una o más veces la cadena A.
A*	Cero o más veces la cadena A.
[abcd]	Alguno de los caracteres que están entre corchetes.
[^abcd]	Cualquier carácter que no esté entre corchetes.
\t	Tabulador.

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:length

“**xs:length**” sirve para especificar una longitud fija.

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:whiteSpace

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.

Se pueden utilizar:

- **preserve** mantiene los espacios en blanco, las tabulaciones, los saltos de línea y los retornos de carro que aparezcan.
- **replace** sirve para sustituir todas las tabulaciones, los saltos de línea y los retornos de carro por espacios en blanco.
- **collapse** se usa 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.

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 mantengan (preserve):

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

Extensiones

“**xs:extensión**” sirve para extender un elemento “**simpleType**” o “**complexType**”.

xs:extension (complexContent)

“**xs:complexContent**”, sirve para definir restricciones o extensiones a un elemento complejo (complexType).

Por ejemplo, dado el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha numero="1">
    <nombre>Eva</nombre>
    <edad>25</edad>
    <ciudad>París</ciudad>
    <pais>Francia</pais>
  </ficha>
  <ficha numero="2">
    <nombre>Giovanni</nombre>
    <edad>26</edad>
    <ciudad>Florencia</ciudad>
    <pais>Italia</pais>
  </ficha>
</fichas>
```

El siguiente archivo “**fichas.xsd**” permitiría validarlo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="fichas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ficha" type="infoPersonaAmpliada"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
```

```

<xs:complexType name="infoPersonaAmpliada">
  <xs:complexContent>
    <xs:extension base="infoPersona">
      <xs:sequence>
        <xs:element name="ciudad" type="xs:string"/>
        <xs:element name="pais" type="xs:string"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="infoPersona">
  <xs:sequence>
    <xs:element name="nombre" type="xs:string"/>
    <xs:element name="edad" type="edadPersona"/>
  </xs:sequence>
  <xs:attribute name="numero" type="xs:integer"/>
</xs:complexType>

<xs:simpleType name="edadPersona">
  <xs:restriction base="xs:integer">
    <xs:minExclusive value="-1"/>
    <xs:maxExclusive value="131"/>
  </xs:restriction>
</xs:simpleType>

</xs:schema>

```

Vemos que, **"infoPersonaAmpliada"** se basa en **"infoPersona"**, añadiéndole dos elementos: "ciudad" y "país".

xs:extension (simpleContent)

xs:simpleContent permite definir restricciones o extensiones a elementos que solo contienen datos, es decir, no contienen a otros elementos.

El siguiente archivo XML:

```

<?xml version="1.0" encoding="UTF-8"?>

<precios xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="precios.xsd">
  <precio moneda="Euro">5</precio>
  <precio moneda="Dólar">6.2</precio>
  <precio moneda="Libra esterlina">4.3</precio>
</precios>

```

Puede validarse con el siguiente documento **"precios.xsd"**:

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="precios">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="precio" maxOccurs="unbounded">
          <xs:complexType>
            <xs:simpleContent>
              <xs:extension base="xs:decimal">

```

```

        <xs:attribute name="moneda" type="xs:string"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Al utilizar **"xs:extensión"**, al elemento **"precio"** hemos incorporado el atributo **"moneda"**.

Documentación del esquema

Veamos ahora el modo de incorporar cierta documentación (quién es el autor, limitaciones de derechos de autor, utilidad del esquema, etc.) a un documento esquema.

Un método para añadir esta información es utilizar comentarios. El problema es que los analizadores no garantizan que los comentarios no se modifiquen al procesar los documentos y por tanto, puede que los datos añadidos se pierdan en algún proceso de transformación del documento.

En lugar de usar los comentarios, XML Schema tiene definido un elemento **"xs:annotation"** que permite guardar información adicional. Este elemento, a su vez puede contener una combinación de otros dos que son:

- **"xs:documentation"**, que además de contener elementos de esquema puede contener elementos XML bien estructurados. También permite determinar el idioma del documento mediante el atributo **"xml:Lang"**.
- **"xs:appinfo"**, se diferencia muy poco del elemento anterior, aunque lo que se pretendió inicialmente era que **"xs:documentation"** fuese legible para los usuarios y que **"xs:appinfo"** guardase información para los programas de software. También es usado para generar una ayuda contextual para cada elemento declarado en el esquema.

Un ejemplo de documentación en un esquema puede ser:

```

<xs:schema xmlns:xs=http://www.w3.org/2001/XMLSchema>
  <xs:annotation>
    <xs:documentation xml:lang="es-es">
      Materiales para formación e-Learning
    <modulo>
      Lenguajes de marcas y sistemas de gestión de información
    </modulo>
    <fecha_creación>2011</fecha_creación>
    <autor>Academia M</autor>
    </xs:documentation>
  </xs:annotation>
  <xs:element name="lmsgi" type="xs:string">
    <xs:annotation>
      <xs:appinfo>
        <texto_de_ayuda>
          Se debe de introducir el nombre completo del tema
        </texto_de_ayuda>
      </xs:appinfo>
    </xs:annotation>
  </xs:element>
</xs:schema>

```