

Bloque2:

Definición de esquemas (XML Schema Document) - XSD



A parte de la siguiente división de **tipos de datos** :

- **PREDEFINIDOS:** Integrados en la especificación de los esquemas XSD.
- **CONSTRUIDOS:** Tipos generados por el usuario basados en tipos predefinidos.

Podemos realizar otra catalogación :

- **Tipos de datos simples (xs:simpleType):** Son los tipos de datos predefinidos anteriores y los tipos de datos contruidos con nuestras propias restricciones (las veremos más adelante).
- **Tipos de datos complejos (xs:complexType):** Son los que permiten definir elementos dentro de otros elementos, los elementos que tienen atributos, los elementos mixtos que tienen datos y otros elementos y los elementos vacíos.

Y dentro de los tipos de datos complejos (**`xs:complexType`**), podemos definir **dos tipos de contenidos** que pueden ir entre etiquetas de apertura y cierre de elementos:

- ❑ **Contenido simple (`xs:simpleContent`)**: Cuando el elemento declarado s lo tienen contenido textual, sin elementos descendientes.
- ❑ **Contenido complejo (`xs:complexContent`)**: Cuando el elemento declarado tiene elementos descendientes, pudiendo tener o no contenido textual.

– M s sobre ELEMENTOS COMPLEJOS, p.e. ELEMENTO VACIO:

Un elemento es complejo (**complexType**) cuando contiene uno o m s elementos y/o atributos. De entre las posibles combinaciones de elementos y/o atributos que puede contener un elemento complejo (1 elemento y 0 atributos, 1 elemento y 1 atributo, 1 elemento y varios atributos, 0 elementos y 1 atributo...) cabe destacar las siguientes:

- **Un elemento complejo puede estar vac o**, es decir, no contener elementos ni texto, pero s  tener al menos un atributo.
- **Un elemento complejo puede contener contenido mixto**, es decir, contener uno o m s elementos, adem s de texto. Por otra parte, podr a tener atributos, o no.

```
<xs:element name="bola">
  <xs:complexType>
    <xs:attribute name="numero" type="numeroDeBola"/>
  </xs:complexType>
</xs:element>

<xs:simpleType name="numeroDeBola">
  <xs:restriction base="xs:positiveInteger">
    <xs:minInclusive value="1"/>
    <xs:maxExclusive value="90"/>
  </xs:restriction>
</xs:simpleType>
```

Ejemplo de elemento "bola" vac o: No puede contener ni otros elementos ni texto, pero s  se le ha incluido un atributo numero

- Permiten establecer **cómo se van a escribir – o utilizar – los elementos de un documento XML.**
- Los principales son:
 - **Indicadores de orden**: secuencia (**sequence**), todo (**all**) y elección (**choice**).
 - **Indicadores de ocurrencia**: **maxOccurs** y **minOccurs**.
 - **Indicadores de grupo**: de elementos (**group**) y de atributos (**attributeGroup**).

<utilizaremos principalmente los de orden y ocurrencia>

Indicadores de orden (**xs:sequence**, **xs:all**, **xs:choice**)

- **xs:sequence** sirve para especificar el orden en el que obligatoriamente deben aparecer los elementos hijo de un elemento
- **xs:all** sirve para indicar que dichos elementos pueden aparecer en cualquier orden.
- **xs:choice** sirve para especificar que solamente se permite escribir uno de los elementos hijo.

Indicadores de orden (xs:sequence, xs:all, xs:choice)

Ejemplo:
"lugar.xsd"

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="lugar">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ciudad">
          <xs:complexType>
            <xs:all>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="pais" type="xs:string"/>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Permite validar
el siguiente XML

```
<?xml version="1.0" encoding="UTF-8"?>
<lugar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="lugar.xsd">
  <ciudad>
    <pais>Italia</pais>
    <nombre>Florencia</nombre>
  </ciudad>
</lugar>
```

Indicadores de ocurrencia (maxOccurs, minOccurs)

- **maxOccurs** y **minOccurs** permiten establecer, el número máximo y mínimo de veces que puede aparecer un determinado elemento. El valor por defecto es 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<países xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="países.xsd">
  <país>
    <nombre>Argentina</nombre>
    <ciudad>Buenos Aires</ciudad>
    <ciudad>Rosario</ciudad>
  </país>
  <país>
    <nombre>México</nombre>
    <ciudad>Guadalajara</ciudad>
    <ciudad>Monterrey</ciudad>
    <ciudad>Cancún</ciudad>
    <ciudad>Mérida</ciudad>
    <ciudad>Ciudad de México</ciudad>
  </país>
  <país>
    <nombre>Colombia</nombre>
  </país>
</países>
```

Ejemplo:

- “país” pueda aparecer una o ilimitadas veces.
- “nombre” tenga que escribirse obligatoriamente, y solo una vez, dentro de “país”.
- De cada “país” puedan escribirse de cero a cinco “ciudades”.

Indicadores de ocurrencia (maxOccurs, minOccurs)

- El código del archivo *“países.xsd”* que validar el XML anterior podría ser:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="países">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pais" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="ciudad" type="xs:string"
                minOccurs="0" maxOccurs="5"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

– Realización del **ejercicio 3** de la actividad práctica 3. AP3. Introducción a XSD:

3. Dado el siguiente documento XML que deberás crear, escribe el contenido del archivo “fichas.xsd” que permita validarlo:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:noNamespaceSchemaLocation="fichas.xsd">
4   <ficha numero="1">
5     <nombre>Ana Sanz Tin</nombre>
6     <edad>22</edad>
7   </ficha>
8   <ficha numero="2">
9     <nombre>Iker Rubio Mol</nombre>
10    <edad>23</edad>
11  </ficha>
12 </fichas>
```