

# RESUMEN XSD

## Asociar un documento XML a un esquema XSD

```
<?xml version = "1.0" encoding = "UTF-8"?>
<clase xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="instituto.xsd">
```

## Definir un esquema XSD

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

## Elementos y atributos

### -Tipo simple (solo datos, sin atributos)

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

#### Atributos

- minOccurs, maxOccurs (ocurrencia mínima y máxima)
- fixed (valor obligatorio)
- default (valor por defecto)
- nullable ("yes" o "no")
- type (xs:string, xs:decimal, xs:integer, xs:boolean, xs:date, xs:time)
- <xs:simpleType> ... </xs:simpleType> (tipos de datos personalizados)
  - <xs:list itemType="xs:integer"/> (lista de valores enteros)
  - <xs:union memberTypes = "xs:integer xs:string"/>
  - <xs:restriction base = "xs:integer" > ... </xs:restriction>
    - <xs:maxInclusive value = "2011"/> (valor máximo incluido 2011)
    - <xs:minInclusive value = "2011"/>
    - <xs:maxExclusive value = "2011"/>
    - <xs:minExclusive value = "2011"/>
    - <xs:minLength value="5"/> (longitud mínima del texto 5 caracteres)
    - <xs:maxLength value="20"/>
    - <xs:enumeration value = "masculino"/> (uno por cada valor posible)
    - <xs:totalDigits value="5"/> (número de dígitos de un número)
    - <xs:fractionDigits value="3"/> (número de decimales de un número)
    - <xs:pattern value = "[0-9]{8} - [A-Z]"/> (patrón expresión regular)

### -Tipo complejo (atributos, elementos, vacíos y mezclados)

#### \*Atributos

Se definen con la etiqueta attribute, exactamente igual que un elemento simple

```
<xs:attribute name = "número" type = "xs:integer" />
```

Atributos posibles: use(required, optional, prohibited), default, fixed

#### \*Elementos con solo texto y atributos

##### Opción 1:

```
<xs:element name="nombre">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string"> (tipo del contenido del elemento)
        <xs:attribute name="cargo" type="xs:string"/> (tipo del atributo)
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

# RESUMEN XSD

Opción 2 (el contenido del elemento solo puede ser texto):

```
<xs:element name="nombre">
  <xs:complexType mixed="true">
    <xs:attribute name="cargo" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

\*Elementos con otros elementos (y opcionalmente atributos)

```
<xs:element name="persona">
  <xs:complexType>
    {sequence, choice, all}
    <xs:attribute name="nombre" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

- <xs:sequence> (elementos, sequence o choice) </xs:sequence>
- <xs:choice> (elementos o sequence para elegir solo 1) </xs:choice>
- <xs:all> (solo elementos sin cardinalidad en cualquier orden) </xs:all>

```
<xs:complexContent> <xs:extension base="tipo"> ... </xs:extension> </xs:complexContent>
(permite crear un complexType a partir de un tipo complejo existente)
```

\*Elementos sin contenido

Sin atributos

```
<xs:element name="cargo" >
  <xs:complexType/>
</xs:element>
```

Con atributos

```
<xs:element name="cargo">
  <xs:complexType>
    <xs:attribute name="antiguedad" use="required" type="xs:gYear"/>
  </xs:complexType>
</xs:element>
```

\*Elementos con contenido mezclado (texto y elementos)

```
<xs:element name="carta" >
  <xs:complexType mixed="true" >
    <xs:sequence >
      <xs:element name="nombre" type="xs:string"/>
      <xs:element name="día" type="xs:gDay"/>
    </xs:sequence >
  </xs:complexType >
</xs:element>
```

## Definiciones fuera de un elemento

### Definición de tipos

Se pueden definir tipos simples o complejos con nombre para reutilizarlos como tipos personalizados en elementos :

```
<xs:simpleType name="dni"> ... </xs:simpleType>
<xs:complexType name="persona"> ... </xs:complexType>
<xs:element name="identificacion" type="dni"/>
<xs:element name="profesor" type="persona"/>
```

### Definición de elementos

Se pueden definir elementos con nombre y hacer referencia a estos con el atributo ref

```
<xs:sequence>
  <xs:element ref="datos"/>
</xs:sequence>
...
<xs:element name="datos"> ... </xs:element name="datos">
```