

XML

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

`<etiquetaPadre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="ficheroxsd.xsd">` para xsd

DTD

Encabezado:

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

`<!DOCTYPE nombre_etiqueta_padre "hoja.dtd">`

`!ELEMENT nombre (hijo1*?|+, hijo2)`

`!ELEMENT hijo1 tipoDato restricción`

`!ATTLIST hijo1 nombreAtributo tipoDeDato restricción`

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

`<!DOCTYPE futbol [`

`<!ELEMENT futbol ((jugador|equipo)*)>`

`<!ELEMENT jugador EMPTY>`

`<!ATTLIST jugador nombre NMTOKENS #REQUIRED>`

`<!ATTLIST jugador codigo ID #REQUIRED>`

`<!ELEMENT equipo EMPTY>`

`<!ATTLIST equipo nombre CDATA #REQUIRED>`

`]>`

Operadores:

`|` (o lógico): o uno o otro

`?`: puede aparecer o no, solo 1 vez

`*`: puede aparecer o no, 1 o varias veces

`+`: tiene que aparecer una o más veces (no puede no aparecer).

Tipos de datos:

`(#PCDATA)`: texto.

`CDATA`: texto pero en los atributos.

`ANY`: cualquier

`EMPTY`: vacío

`NMTOKEN`: letras, dígitos, y los caracteres punto ".", guion "-", subrayado "_" y dos puntos ":".

`NMTOKENS`: como el tipo `NMTOKEN` y **también espacios en blanco**.

`ID`: el valor del atributo (no el nombre) debe ser **único y no se puede repetir**.

IDREF: referencia a atributo ID de otro elemento

IDREFS: serie de valores separados por espacios que coinciden con el valor del atributo ID de otros elementos.

ENTITY: entidad definida en la DTD.

ENTITIES: lista de entidades definida en la DTD.

NOTATION: notación definida en la DTD.

Restricciones de atributos y de elementos:

- **#REQUIRED**: obligatorio.
- **#FIXED** valor: valor fijo
- **#IMPLIED**: opcional
- (valor1|valor2): valor o valores predeterminado. No se pone el tipo de dato, ya va implícito en los valores.

XSD

Encabezado:

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

Encabezado en el documento xml dentro de la etiqueta padre:

```
<nombre_etiqueta_padre xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="definicion.xsd">
```

Etiquetas:

-xs:element: name = "Nombre" type = "Tipo complejo si se define"

-xs:simpleType

-xs:restriction base: "tipo de dato" - Ojo, si le pones restricción, el tipo de dato va aquí, no en el elemento

-xs:min-Inclusive: value = "valor mínimo"

-xs:max-inclusive: value = "valor máximo"

-xs:enumeration: value= "valor único" para delimitar los datos a introducir.

-xs:pattern: value="[a-zA-Z0-9]{8}"

-xs:attribute: name = "Nombre" type = "Tipo complejo si se define"

```
<xs:element name="atributo">
  <xs:simpleType>
    <xs:restriction base="tipo de dato">
      <xs:pattern value="\d{2}\.\d{3}\.\d{3}-[A-Za-z]"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```
</xs:restriction>
</xs:simpleType>
</xs:element>
```

-xs:simpleType

-xs:restriction base: “tipo de dato”

-xs:min-Inclusive: value = “valor mínimo”

-xs:max-inclusive: value = “valor máximo”

-xs:enumeration: value= “valor único” para delimitar los datos a introducir.

-xs:pattern: value= “[0-9]”

-xs:element: name = “Nombre” **COMPUESTO**

-xs:complexType name= “Nombre del tipo complejo”

-xs:sequence

-xs:element:

```
<xs:complexType name="etiquetapadre">
  <xs:sequence>
    <xs:element name="etiqueta1" type="tipo_Dato"/>
    <xs:element name="etiqueta2" type="tipo_Dato"/>
    <xs:element name="etiqueta3" type="tipo_Dato"/>
  </xs:sequence>
  <xs:attribute name="provincia" type="xs:NMTOKEN" fixed="Lugo"/>
</xs:complexType>
```

Tipos de datos:

xs:string
xs:decimal
xs:integer
xs:boolean
xs:date
xs:time

Opciones de atributos y elementos:

name= “nombre del elemento”

type= “Tipo de dato”

minOccurs= “elementos mínimos”

maxOccurs= “elementos máximos o unbounded”

default="dato por defecto".

fixed="Dato fijado"

use="required" - Dato requerido.

```
<!-- definition of simple elements -->
<xs:element name="orderperson" type="xs:string"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="address" type="xs:string"/

<!-- definition of attributes -->
<xs:attribute name="orderid" type="xs:string"/>

<!-- definition of complex elements -->
<xs:element name="shipto">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="name"/>
      <xs:element ref="address"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```