

Instituto FOC
Módulo de desarrollo de aplicaciones web
Lenguajes de marcas y sistemas

BRUNO MARENCO CERQUEIRA

Tarea Colaborativa: Intercambio de datos

Mayo/2018

Índice

Índice.....	2
Participación y organización.....	3
Planteamiento	3
Generar 2 ficheros (CLIENTES.XML y CLIENTES.XSD) Realizado por Bruno Marengo Cerqueira.....	3
XML	4
XSD	5
Validación al XML	7
Validación al XSD	7
Validación al XML con el XSD	8
Generar 2 ficheros (ARTICULOS.XML y ARTICULOS.XSD	8

Participación y organización

El trabajo se divide en diferentes apartados:

- Creación y validación XML.
- Creación y validación XSD.
- Validación del XML con el XSD.

Cada uno de los apartados se divide en dos, siendo responsabilidad de un alumno cada parte del trabajo. La generación de archivos Clientes y Artículos.

De esta forma, los archivos sobre los Clientes será realizado por Bruno Marengo Cerqueira, como fue comunicado a través del foro de la tarea, quedando los archivos de los artículos a cargo del otro integrante. El otro integrante no ha respondido por el chat nada, por lo que no ha entregado su parte y el trabajo está incompleto.

Planteamiento

Dos empresas, que están colaborando de forma empresarial, han decidido dar un paso más en su relación intercambiando datos de sus clientes y productos para hacerse más competitivos y crecer como grupo empresarial.

Entre la información que quieren intercambiar esta lo siguiente:

Generar 2 ficheros (CLIENTES.XML y CLIENTES.XSD) Realizado por Bruno Marengo Cerqueira.

- Datos de clientes:
 - Nombre: Cadena de texto
 - Nif: Cadena de texto
 - Tipo: Persona física o persona jurídica
 - Años de relación: Numérico positivo.
 - Artículos habituales:
 - Id de artículo: Numérico
 - Pedido medio: Numérico

XML.

El archivo XML lo utilizamos para intercambiar los datos de los clientes entre las dos empresas, primero indicamos la instrucción de procesamiento y con el tag raíz clientes lo referenciamos al archivo XSD. Y al final del documento cerramos el tag clientes.

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

Los datos clientes los incluimos dentro del tag clientes. Uno de estos datos, el NIF lo incluimos como atributo de cliente.

```
<cliente nif="91820114f">
...
</cliente>
```

Dentro de este tag clientes colocamos los diversos elementos que componen la información: nombre, tipo, años de relación.

```
<nombre>Antonio Torres Yanes</nombre>
<tipo>persona física</tipo>
<años_de_relacion>10</años_de_relacion>
```

Y también colocamos otro elemento, dentro del cual tenemos otros dos elementos uno dentro de otro. Dentro de artículos habituales colocamos el id del artículo y dentro de este el elemento de pedido medio.

```
<articulos_habituales>
  <articulo_id>561<pedido_medio>14</pedido_medio></articulo_id>
  <articulo_id>43<pedido_medio>10</pedido_medio></articulo_id>
</articulos_habituales>
```

Esta ha sido la estructura elegida para el XML, siguiendo lo exigido por el XSD, que se detalla a continuación, para su validación.

XSD.

Primeramente, colocamos la instrucción de procesamiento y hacemos que cada uno de los elementos del esquema empiece con `<xs:>`, asociado con el espacio de nombres del Esquema XML a través de la segunda línea.

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

Creamos un tipo complejo para el elemento inicial, después del root y que engloba todas las informaciones inferiores, el elemento cliente. Para ello, lo definimos como tipo complejo (`<xs:complexType>`) y creamos una restricción por la que pueden existir infinitud de tags cliente (`<xs:element name="cliente" maxOccurs="unbounded"/>`) dentro de (`<xs:sequence>`), que sirve para indicar los elementos dentro de otro elemento o colocar restricciones.

```
<xs:element name="clientes">
<xs:complexType>
  <xs:sequence>
    <xs:element name="cliente" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
```

Abrimos el elemento nuevamente con otra secuencia para poder incluir los elementos que forman parte de cliente.

```
<xs:element name="cliente" >
  <xs:complexType>
    <xs:sequence>
      ....
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

El tag clientes tiene un atributo que se especifica después de la secuencia de elementos, que guarda la información del nif del cliente.

```
<xs:attribute name="nif" type="xs:string"/>
```

Dentro de la secuencia de cliente tenemos 3 elementos simples. Dos declarados en una línea indicando el nombre y el tipo (string para caracteres y positiveInteger para un número positivo) que son:

```
<xs:element name="nombre" type="xs:string"/>
<xs:element name="años_de_relacion" type="xs:positiveInteger"/>
```

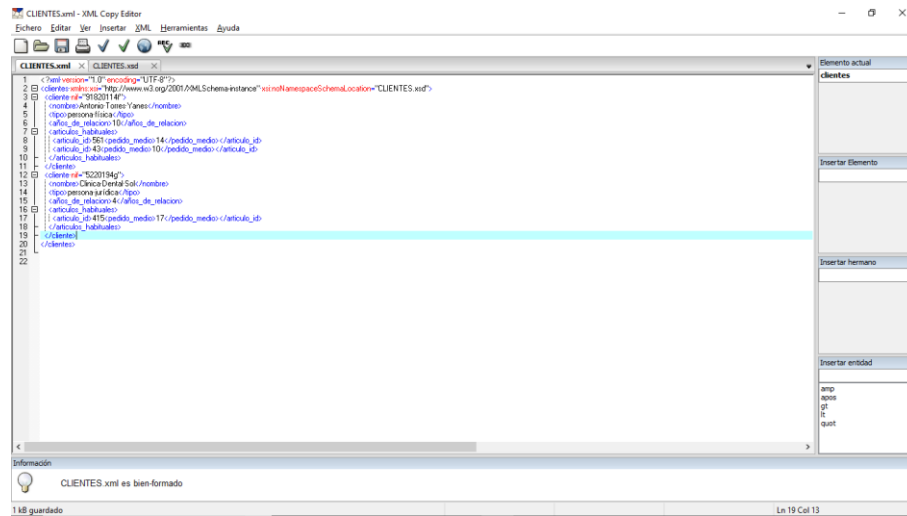
Y el tercer elemento simple que es el tipo que es un string, el cual abrimos para incluir un número de valores que son permitidos dentro de este elemento con la declaración (<xs:enumeration value="valor permitido"/>).

```
<xs:element name="tipo">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="persona física"/>
      <xs:enumeration value="persona jurídica"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

El último elemento dentro de cliente es un elemento de tipo complejo, artículos habituales, el cual tiene otro tipo complejo dentro, el id del artículo y este último tiene un elemento simple dentro, pedido medio, aunque también acepta texto, por lo que es un tipo complejo mixto.

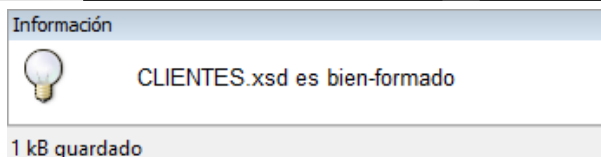
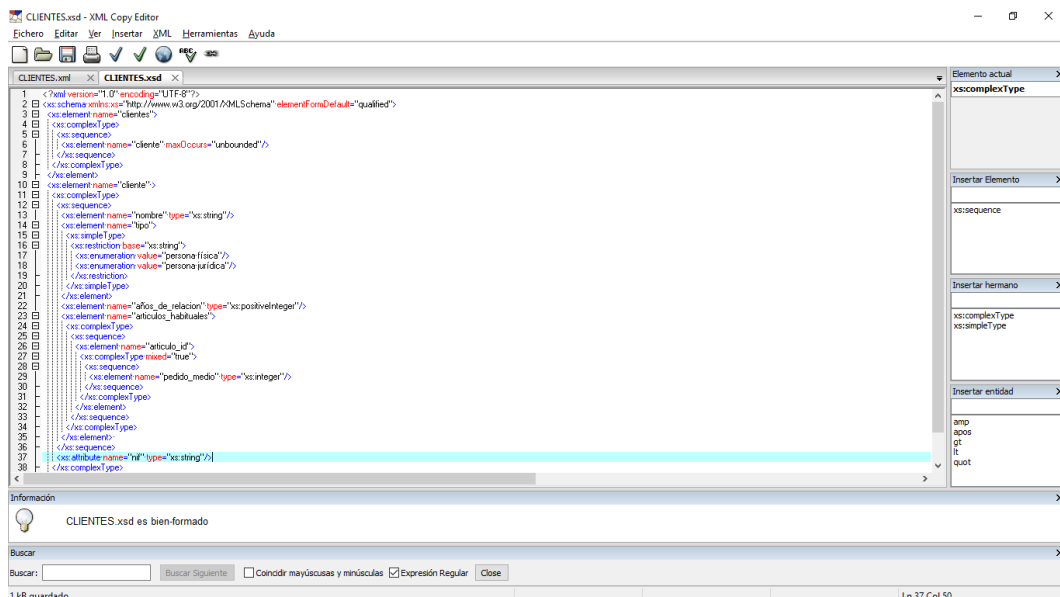
```
<xs:element name="articulos_habituales">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="articulo_id">
        <xs:complexType mixed="true">
          <xs:sequence>
            <xs:element name="pedido_medio" type="xs:integer"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

Validación al XML.



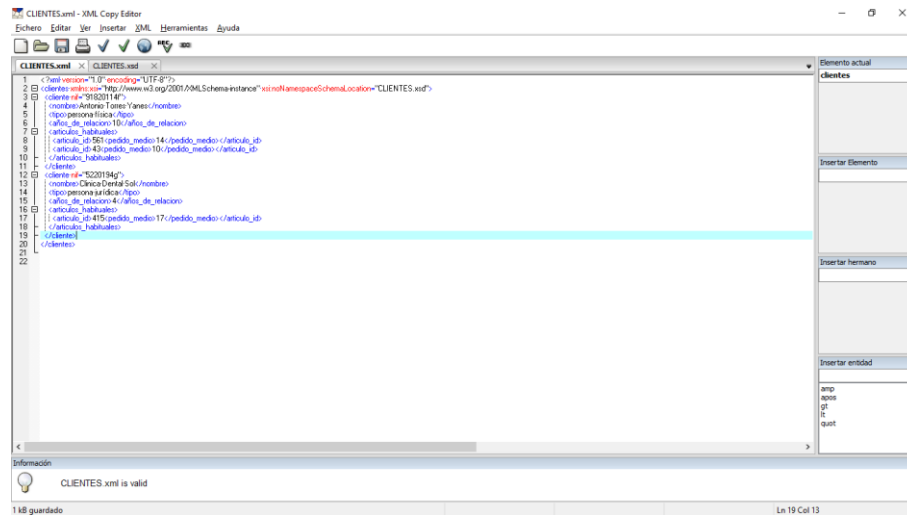
Se puede comprobar en las capturas de encima que el archivo CLIENTES.xml está bien formado.

Validación al XSD.

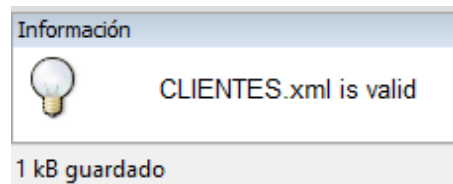


Se puede comprobar en las capturas de encima que el archivo CLIENTES.xsd está bien formado.

Validación al XML con el XSD.



Zoom.



Se puede comprobar en las capturas de encima que el archivo CLIENTES.xml es validado por el archivo CLIENTES.xsd.

Generar 2 ficheros (ARTICULOS.XML y ARTICULOS.XSD)

- Artículos:
 - Id de artículo: numérico
 - Nombre: Cadena de texto
 - Tipo: Electrónica o informática.
 - Stock: Numérico
 - Proveedor:
 - Id de proveedor: Numérico
 - Nombre: Cadena de texto