

Teoría XSD

Índice

1	¿Qué es un XSD?	2
2	¿Cómo referenciar un XSD en un XML?	2
2.1	Asociado a un espacio de nombres	2
2.2	Sin asociar a un espacio de nombres	4
3	Definición de elementos simples	5
4	Definición de elementos complejos	5
5	Definición de atributos	6
6	Tipos de datos	6
7	Restricciones sobre los datos	6
8	Referencias especiales	7
9	Webs de consulta y Webgrafía	7

1 ¿Qué es un XSD?

En el siguiente enlace podéis encontrar una breve presentación e introducción a lo que son los XML Schema Definition (de aquí en adelante XSD).

https://www.w3schools.com/xml/schema_intro.asp

Básicamente un XSD es un documento que define la estructura que debe tener un documento XML, de forma similar a como lo hacíamos con DTD pero con mucha más flexibilidad que este último. El procesador XML utiliza el XSD para verificar si un documento es válido, es decir, si el documento cumple las reglas del XSD.

Mediante XSD podemos definir los siguientes aspectos de un documento XML:

- Los elementos y atributos que pueden aparecer en el documento XML.
- El número exacto y el orden en el que deben aparecer los elemento hijos de un elemento concreto.
- Los tipos de datos que contienen tanto los elementos finales como los atributos.
- Valores por defecto o fijos tanto para los elementos finales como para los atributos.

2 ¿Cómo referenciar un XSD en un XML?

En el caso de los XSD son documentos siempre externos al propio documento XML, no como en los DTD que admitían la opción de ser declarados de forma interna. Sin embargo sí que disponemos de dos opciones diferentes a la hora de referenciar el documento.

2.1 Asociado a un espacio de nombres

Si asociamos el XSD a un espacio de nombres concreto nos veremos obligados a incluir dentro del elemento raíz varios atributos,

tanto en el XML como en el XSD. Dichos atributos son los siguientes.

Dentro del XML:

- **xmlns.** Especifica cual será el espacio de nombres por defecto.
- **xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance".** Es obligatorio siempre que se valide contra un XSD.
- **xsi:schemaLocation.** Define dos cosas (ambos valores deben estar separados por un espacio en blanco), el espacio de nombres por defecto usado y la ubicación del archivo que contiene el XSD.

Por lo tanto la definición del primer elemento quedaría de la siguiente forma:

<elementoRaiz

xmlns="EspacioDeNombresPorDefecto"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="EspacioDeNombresPorDefecto ArchivoXSD">

Dentro del XSD:

- **xmlns:xs="http://www.w3.org/2001/XMLSchema".** Es obligatorio en todo documento XSD. (En alguna documentación podéis encontrar que se utilizar el prefijo “xsd” en lugar de “xs”, lo único importante es que se utilice el mismo en todo el documento XSD).
- **targetNamespace.** Es donde se indica el espacio de nombres definido para este esquema. Se puede usar el de por defecto.
- **xmlns.** Es donde se indica el espacio de nombres por defecto usado.
- **elementFormDefault="qualified".** Indica que todos los elementos usados por el XML que se declararon en este esquema deben estar calificados para el espacio de nombres.

Por lo tanto la definición del elemento “**schema**” (que es siempre el primer elemento en todo documento XSD) quedaría de la siguiente forma:

```
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="EspacioDeNombresDefinidoParaElEsquema"
xmlns="EspacioDeNombresPorDefecto"
elementFormDefault="qualified">
```

2.2 Sin asociar a un espacio de nombres

Si no asociamos el XSD a un espacio de nombres concreto la definición de las cabeceras se reduce sensiblemente, pero aun así nos veremos obligados a incluir dentro del elemento raíz varios atributos, tanto en el XML como en el XSD. Dichos atributos son los siguientes.

Dentro del XML:

- **xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"**. Es obligatorio siempre que se valide contra un XSD.
- **xs:noNamespaceSchemaLocation**. Indica la ubicación del archivo que contiene el XSD.

Por lo tanto la definición del primer elemento quedaría de la siguiente forma:

```
<elementoRaiz
xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
xs:noNamespaceSchemaLocation="ArchivoXSD">
```

Dentro del XSD:

- **xmlns:xs="http://www.w3.org/2001/XMLSchema"**. Es obligatorio en todo documento XSD. (En alguna documentación podéis encontrar que se utilizar el prefijo “xsd” en lugar de “xs”, lo único importante es que se utilice el mismo en todo el documento XSD).

Por lo tanto la definición del elemento “**schema**” (que es siempre el primer elemento en todo documento XSD) quedaría de la siguiente forma:

```
<xs:schema
```

```
xmlns:xs="http://www.w3.org/2001/XMLSchema">
```

Para la mayor parte de los ejercicios de esta parte del temario vamos a emplear la segunda formula que es más sencilla y nos va a resultar suficiente, pero es importante conocer ambas opciones.

3 Definición de elementos simples

Los elemento simples (o finales) son aquellos que solo contienen datos en el XML.

En el siguiente enlace podéis consultar como se define un elemento simple.

https://www.w3schools.com/xml/schema_simple.asp

4 Definición de elementos complejos

Se entiende por elemento complejo a todo aquel elemento de un documento XML que cumple alguna de las siguientes características:

- **El elemento tiene algún atributo**, independientemente de si es un elemento vacío o además contiene texto.
- **El elemento contiene a otros elementos**, independientemente de si tiene o no atributos.

En el siguiente enlace podéis consultar como se definen los elementos complejos.

https://www.w3schools.com/xml/schema_complex.asp

5 Definición de atributos

Los atributos solo pueden pertenecer a elementos complejos, ya que la mera presencia de un atributo convierte al elemento en complejo. La forma de declarar los atributos la podéis consultar en el siguiente enlace.

https://www.w3schools.com/xml/schema_simple_attributes.asp

A la hora de declarar los atributos hay que tener en cuenta que es muy importante la posición de la declaración dentro del elemento complejo. Los atributos hay que declararlos siempre al final, es decir, antes de cerrar el “**complexType**” y fuera del “**sequence**” en el caso de los elementos que contengan a otros elementos.

6 Tipos de datos

Los diferentes tipos de datos disponibles para los elementos simples y para los atributos podéis consultarlos en la siguiente tabla.

Datos tipo	Significado
Texto	https://www.w3schools.com/xml/schema_dtypes_string.asp
Fecha y Hora	https://www.w3schools.com/xml/schema_dtypes_date.asp
Numéricos	https://www.w3schools.com/xml/schema_dtypes_numeric.asp
Variados	https://www.w3schools.com/xml/schema_dtypes_misc.asp

7 Restricciones sobre los datos

En el siguiente enlace podéis consultar las restricciones que se pueden aplicar a los valores de los datos contenidos en los elementos o atributos en XSD.

https://www.w3schools.com/xml/schema_facets.asp

8 Referencias especiales

En XSD hay unas etiquetas que nos permiten aplicar una serie de reglas concretas, entre las más destacadas están:

- Establecer el orden en el que aparecen los elementos hijos mediante **“sequence”** y **“all”**.
- Crear nuevos tipos de datos mediante **“complexType”** y **“simpleType”**.
- Crear nuevos tipos de contenido mediante **“complexContent”** y **“simpleContent”**.
- Crear restricciones y extensiones de los datos definidos por el usuario mediante **“restriction”** y **“extension”** respectivamente.

En el siguiente enlace podéis consultar todas las etiquetas que se pueden usar en XSD, además de su definición y sobre todo dentro de que otras etiquetas se pueden emplear (esto último es muy importante).

https://www.w3schools.com/xml/schema_elements_ref.asp

9 Webs de consulta y Webgrafía

La web principal en la que nos vamos a basar (y de la que he extraído estos apuntes) es la siguiente:

W3Schools. XML Schema Tutorial

https://www.w3schools.com/xml/schema_intro.asp