

UF3 · Actividad 2

Tarea en equipo (5 personas)

XML, DTD y XSD

MIGUEL ÁNGEL DEL RINCÓN MANRIQUE
JOSÉ ANTONIO SANTOS SÁNCHEZ
MIGUEL ÁNGEL LOZANO BERMEJO
ALVARO QUILES POMARES
RAÚL RUIZ SANZ



MP_0373
Lenguajes de
marcas y sistemas
de gestión

INDICE

REPOSITORIO GITHUB.....	4
PUESTA EN COMÚN.....	4
ARCHIVO XML.....	5
ARCHIVO DTD.....	6
ARCHIVO XSD.....	6
VALIDADORES ON-LINE.....	7

Actividad 2. Tarea en equipo (5 personas). XML, DTD y XSD

Enunciado

Vamos a elaborar una estructura de base de datos en **XML** que permita almacenar los datos de una biblioteca en la red. Para ello se pide que elabores un DTD que permita validar los documentos XML con las siguientes características:

- Existen tres tipos de documentos almacenados en la biblioteca: libros, revistas y periódicos. Todos los documentos están identificados por el atributo Id.
- Para los libros este atributo empieza con la letra "L" seguido de 4 dígitos identificativos.
- Para los periódicos este atributo empieza por la letra "P" seguido de los 4 dígitos identificativos.
- En el caso de las revistas empieza por la letra "R".
- Los libros a su vez son clasificados en novela, infantil o didáctico. Cada libro contiene un atributo identificativo de su clase denominado tipo_clase. Dentro de cada libro se tiene un título, varios capítulos con el título en su interior, un índice y una sinopsis. Tanto en libro como en capítulo existe un atributo que contiene el número de páginas del libro o del capítulo, según corresponda. Cada capítulo contiene un elemento denominado contenido, en el que se tiene un atributo con el enlace a la información.
- Las revistas a su vez son clasificadas en: informática, corazón, coches, investigación y otras. Cada revista tiene el atributo tipo_clase identificativo de la clase a la que pertenece. Dentro de cada revista tenemos el título, el número de la revista, un índice de contenido y las secciones. En cada sección y en la revista se tiene un atributo que contiene el número de páginas. Además, en cada sección se tiene la parte denominada contenido, en la que se tiene un atributo con un enlace a la información. De igual forma que en los anteriores, la última etiqueta del árbol debe ser el contenido con un atributo que referencia al contenido.
- Los periódicos se clasifican en nacionales e internacionales. Cada periódico contiene el atributo tipo_clase identificativo de la clase a la que pertenece y un atributo que incluye la fecha de publicación.
- Dentro de los periódicos tenemos secciones y un índice. Cada sección debe contener un atributo identificativo del tipo de sección, que puede ser: económica, opinión, deportes, nacional o internacional. Las secciones se dividen en artículos, en donde se define en un atributo el autor. Finalmente, el contenido será el último elemento del árbol, que necesita un atributo que referencia a la información.

Requerimiento 1

Crear un **XML** con el modelo de datos indicado en el enunciado.

Elaborar un DTD que permita validar el documento XML.

Validar el documento con alguna aplicación externa.

IMPORTANTE: Cada alumno propondrá una solución de XML y DTD. Una vez todos los alumnos hayan puesto todas sus soluciones, se cotejará con el resto para poner la solución final.

Requerimiento 2

Elaborar un **XSD** que permita validar el documento XML.

Validar el documento con alguna aplicación externa.

IMPORTANTE: Cada alumno propondrá una solución de XML y XSD. Una vez todos los alumnos hayan puesto todas sus soluciones, se cotejará con el resto para poner la solución final.

Detalles de entrega

El código fuente estará alojado en un repositorio GITHUB.

» REPOSITORIO GITHUB

GitHub:

<https://github.com/MiguelALozano/LMactividad2.git>

» PUESTA EN COMÚN

Cada miembro del grupo ha realizado sus códigos y los hemos puesto en común en diferentes sesiones conjuntas.

Varias de las propuestas coincidían en utilizar como etiquetas documentos con atributo id, el cual filtraría que tipo de documento sería (libro, revista o periódico), pero se opta por utilizar como etiqueta raíz *<biblioteca>* la cual tendrá tres hijos, *<libros>*, *<revistas>* y *<periodicos>*, cada una con sus correspondientes clasificaciones y atributos.

Dentro de *<libros>* podremos insertar una cantidad indeterminada de elementos *<libro>*, de igual manera dentro de *<revistas>* podremos insertar *<revista>* y dentro de *<periodicos>* haremos lo mismo con *<periodico>*.

Se decide realizar tanto el DTD como el XSD en archivos externos, para un mejor control de validación, aunque para la utilización de alguna aplicación externa, hemos tenido que incrustar el código DTD en el archivo XML, pues sólo admitía esta opción.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<biblioteca libro_id="L0001" tipo_clase="novela" numero_paginas="205">
  <titulo>Patata</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="20">
      <contenido_libro enlace="https://www.patata.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>Una Patata muy maja.</sinopsis>
</libro>

<libro id="L0002" tipo_clase="infantil" numero_paginas="350">
  <titulo>Patatín</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="10">
      <contenido_libro enlace="https://www.patatin.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>La hija de Patata.</sinopsis>
</libro>

<libro id="L0003" tipo_clase="didactico" numero_paginas="700">
  <titulo>Enciclopedia de la Patata.</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="30">
      <contenido_libro enlace="https://www.wikipatata.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>La enciclopedia preferida de estas dos.</sinopsis>
```

» ARCHIVO XML

Creamos dos archivos XML para realizar las dos llamadas a los archivos externos de validación DTD y XSD.

Los hemos codificado con *encoding UTF-8* y referenciado la localización de los archivos DTD y XSD. En el siguiente código vemos la llamada al archivo .xsd.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<biblioteca xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="biblioteca.xsd">
```

La estructura pensada para el elemento **<libro>** es la siguiente:

- Atributos *id* (como ID), *tipo_clase* (como Lista) y *numero_paginas* (CDATA).
- Tendrá los hijos **<titulo>**, **<indice>**, **<capitulos>** y **<sinopsis>**.
- El elemento **<capitulos>** tendrá un elemento hijo llamado **<capitulo>** con un atributo *numero_paginas* (CDATA).
- El elemento **<capitulo>** contiene un hijo, **<contenido_libro>**, con un atributo llamado *enlace* (CDATA).

La estructura pensada para el elemento **<revista>** es la siguiente:

- Atributos *id* (como ID), *tipo_clase* (como Lista) y *numero_paginas* (CDATA).
- Tendrá los hijos **<titulo>**, **<numero_revista>**, **<indice>** y **<secciones>**.
- El elemento **<secciones>** contiene un hijo, **<seccion_revista>** con un atributo *numero_paginas* (CDATA).
- El elemento **<seccion_revista>** contiene un hijo, **<contenido_revista>** con un atributo llamado *enlace* (CDATA).

La estructura pensada para el elemento **<periodico>** es la siguiente:

- Atributos *id* (como ID), *tipo_clase* (como Lista) y *fecha_publicacion* (CDATA).
- Tendrá los hijos **<indice>**, y **<secciones>**.
- El elemento **<secciones>** tendrá un elemento hijo llamado **<seccion_periodico>** con un atributo *tipo_seccion_periodico* (tipo Lista y obligatoria).
- El elemento **<seccion_periodico>** contiene un hijo, **<articulos>**, y este a su vez contiene un hijo llamado **<articulo>** que tiene un atributo llamado *autor* y un hijo llamado **<contenido_articulo>**.
- El elemento **<contenido_articulo>** tiene un atributo llamado *enlace* el cual contiene una URL al elemento externo.

```
<sinopsis>La hija de Patata.</sinopsis>
</libro>

<libro id="L0003" tipo_clase="didactico" numero_paginas="700">
  <titulo>Enciclopedia de la Patata.</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="30">
      <contenido_libro enlace="https://www.wikipatata.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>La enciclopedia preferida de estas dos.</sinopsis>
```

» ARCHIVO DTD

Creamos la estructura *biblioteca.dtd* para la validación del documento XML.

En el ejercicio se indica que los ID's de Libro, revista y periódico tienen que seguir un patrón definido y hemos comprobado que con DTD no es posible, ya que no acepta la validación de la estructura requerida, letra inicial seguido de 4 dígitos identificativos.

Al no poder utilizar el patrón requerido en el ejercicio optamos por usar el atributo con tipo ID, ya que nos asegura que el código sea único.

También hemos decidido tratar los atributos como obligatorios, añadiendo #REQUIRED, excepto en los de tipo lista, en los que hemos añadido un valor por defecto. Para más detalle en el propio archivo hemos dejado comentado todo el código con las explicaciones pertinentes.

» ARCHIVO XSD

Creamos la estructura *biblioteca.xsd* en la cual hemos podido realizar la verificación de toda la estructura del documento al igual que en el DTD, así como en aquellos casos que tenemos una lista de valores específicos como únicos válidos.

Adicionalmente, solventando el inconveniente del DTD, al ser posible el uso de patrones, hemos podido validar el id con su letra correspondiente (L, R, P) seguido de los 4 dígitos identificativos.

De manera análoga hemos verificado la existencia de todos los elementos y atributos, que además deben ir en un orden concreto.

Para los atributos *tipo_clase* se les ha colocado un valor *default* en vez de hacerlos *required*.

Para más detalle en el propio archivo hemos dejado comentado todo el código con las explicaciones pertinentes.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<libro id="L0001" tipo_clase="novela" numero_paginas="205">
  <titulo>Patata</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="20">
      <contenido_libro enlace="https://www.patata.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>Una Patata muy maja.</sinopsis>
</libro>

<libro id="L0002" tipo_clase="infantil" numero_paginas="350">
  <titulo>Patatín</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="10">
      <contenido_libro enlace="https://www.patatin.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>La hija de Patata.</sinopsis>
</libro>

<libro id="L0003" tipo_clase="didactico" numero_paginas="700">
  <titulo>Enciclopedia de la Patata.</titulo>
  <indice>Índice</indice>
  <capitulos>
    <capitulo numero_paginas="30">
      <contenido_libro enlace="https://www.wikipatata.com"></contenido_libro>
    </capitulo>
  </capitulos>
  <sinopsis>La enciclopedia preferida de estas dos.</sinopsis>
</libro>
```

» VALIDADORES ON-LINE

Además de la validación propia del programa Visual Studio Code, hemos utilizado los siguientes validadores on-line:

www.w3schools.com/xml/xml_validator.asp

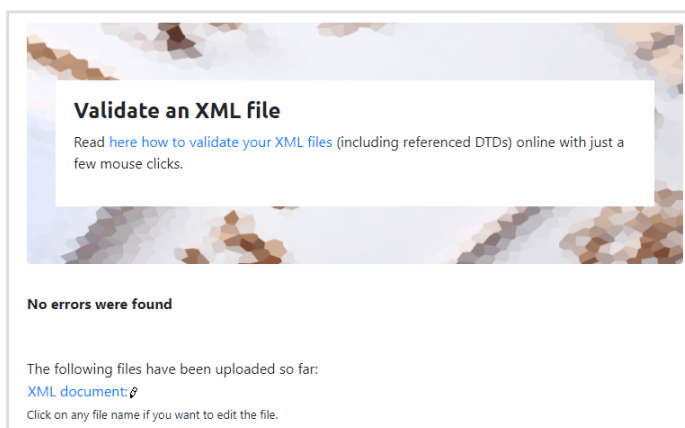
Admite un sólo archivo, en el cual el DTD debe ir embebido en el archivo XML.



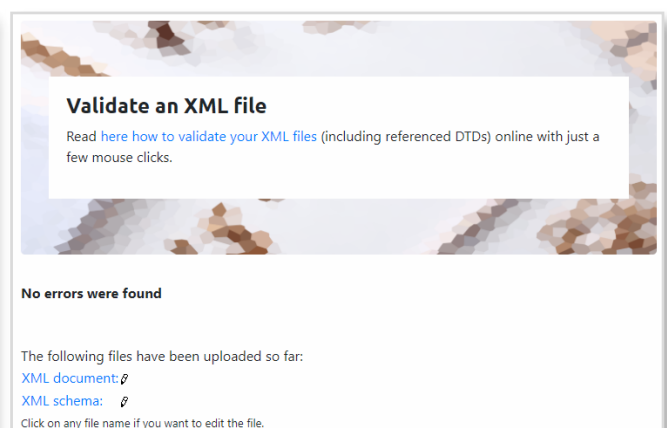
Validación con DTD.

<http://www.xmlvalidation.com>

En este validador tenemos la opción de usar archivos independientes, tanto para el DTD como para el XSD.



Validación con DTD embebido.



Validación con XSD externo.

<http://xmlvalidator.new-studio.org/>

En esta web, para la validación con Schema, tenemos la opción de usar un archivo externo y para validar con DTD la única opción que nos permite usar es con el código embebido en el XML.

En las siguientes imágenes podemos observar tanto el resultado de validación con Schema como el resultado de validación con DTD.

XML Validator Online BETA

XML Validation

☐ XPath ☒ Schema ☐ DTD

Charset: UTF-8 (Unicode, worldwide)

Choose XML file to validate: bibliotecaXSDexterno.xml

XML Schema: biblioteca.xsd

XPath Explorer

Validation Result

✓ Validation successful!

Source Code

```

1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <biblioteca xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="biblioteca.xsd">
3.   <!-- elemento raíz -->
4.   <!-- Dentro del elemento raíz, se crean las ramas libros, revistas y periódicos que serán las contenedoras de cada libro, revista y
5.   -->
6.   <!-- ZONA LIBROS recoge todos los libros que contiene la biblioteca -->
7.   <libros>
8.     <libro id="10001" numero_paginas="200">
9.       <titulo>Patata</titulo>
10.      <indice>indice</indice>
11.      <capitulo>
12.        <capitulo numero_paginas="20">
13.          <contenido_libro enlace="https://www.patata.com"></contenido_libro>
14.        </capitulo>
15.      </capitulo>
16.      <sinopsis>Una Patata muy maja.</sinopsis>
17.    </libro>
18.    <libro id="10002" tipo_clase="infantil" numero_paginas="350">
19.      <titulo>Patata</titulo>
20.      <indice>indice</indice>
21.      <capitulo>
22.        <capitulo numero_paginas="10">
23.          <contenido_libro enlace="https://www.patata.com"></contenido_libro>

```

Validación con Schema.

XML Validator Online BETA

XML Validation

☐ XPath ☐ Schema ☒ DTD

Charset: UTF-8 (Unicode, worldwide)

Choose XML file to validate: bibliotecaDTDembebido.xml

Due to the technical issue, embedded DTD is allowed ONLY.

XPath Explorer

Validation Result

✓ Validation successful!

Source Code

```

1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <!DOCTYPE biblioteca [
3.   <ELEMENT biblioteca (libros, revistas, periodicos)>
4.   <!-- Tipos de documentos almacenados en la biblioteca -->
5. ]>
6.
7. <!-- ZONA LIBROS -->
8. <ELEMENT libros (libro)*>
9. <!-- Decidimos que de este elemento podemos tener entre 0 y muchos -->
10. <ELEMENT libro (titulo, indice, capitulos, sinopsis)>
11. <!-- Elementos hijos de libro -->
12. <ATTLIST libro
13.   id ID #REQUIRED
14.   tipo_clase (novela|infantil|didactico) "novela"
15.   numero_paginas CDATA #REQUIRED
16. ]>
17.
18. <!-- Atributos de libro, id de la c-->
19. <ELEMENT capitulos (capitulo)*>
20. <!-- capitulos contiene el nombre 1 capitulo -->
21. <ELEMENT capitulo (contenido_libro)>
22. <!-- A su vez cada capitulo tiene un elemento hijo que es el contenido del capitulo-->
23. <ATTLIST capitulo numero_paginas CDATA #REQUIRED
24. <!-- El elemento contenido_libro se declara como vacío, ya que no tiene ningún texto ni elemento en su interior.
25. A dicho contenido se accede a través del atributo enlace -->

```

Validación con DTD.

<https://www.freeformatter.com/xml-validator-xsd.html>

Esta web, sólo podemos usarla con Schema, tenemos la opción de subir los dos archivos o copiar el código y pegarlo directamente en la web.

The XML document is valid.

XML Input

Option 1: Copy-paste your XML document here

Option 2: Or upload your XML document

Ningún archivo seleccionado UTF-8

XSD Input (Optional if XSD referred in XML using schemaLocation)

Option 1: Copy-paste your XSD document here

Option 2: Or upload your XSD document

Ningún archivo seleccionado UTF-8

Validación con XSD como archivo externo.