Nota: en estas actividades es recomendable separar cada apartado en un directorio distinto, para evitar confusión con los nombres de ficheros.

Actividad 13. Vincular XML a XSD

13.1 Sin espacio de nombres

Una aplicación recibe datos de alumnos y los introduce de forma automática en una base de datos. Por cada alumno matriculado, recibe un documento xml con los datos del alumno. Para comprobar que los datos se reciben en el formato adecuado, se define el esquema datosPersonales.xsd.

Modifica el fichero act13_1.xml para que se valide con el esquema datosPersonales.xsd. No debe utilizarse ningún espacio de nombres.

Comprueba que el xml valida correctamente.

13.2 Con espacio de nombres sin calificar

Modifica los ficheros del apartado anterior para que se utilice un espacio de nombres con el prefijo “dp” (datos personales) en el xml. Es decir, en lugar de <alumno> se emplee <dp:alumno>.

Por defecto los elementos deben estar **no calificados** (añade elementFormDefault="unqualified" al esquema).

Comprueba que el xml valida correctamente.

13.3 Con espacio de nombres calificado

Modifica los ficheros del apartado anterior para que por defecto los elementos estén **calificados**. Es decir, todos requieren el prefijo “dp”.

Comprueba que el xml valida correctamente.

Actividad 14. Vincular XML a XSD con varios espacios de nombres

14.1 Validación con dos esquemas calificados

Descarga el fichero act14_1.zip.

Abre el fichero act14_1.xml. Observa que tiene dos prefijos distintos en los elementos xml. Observa también que hay nombres de elementos duplicados, pero con prefijos distintos (<dirección>).

Lo que se pretende es que la validación de este documento se realice empleando **dos** esquemas distintos:

- Los datos relativos al alumno tienen prefijo “dp” (datos personales) y se validarán con el esquema datosPersonales.xsd.
- Los datos relativos al centro educativo tienen prefijo “ce” (centro educativo) y se validarán con el esquema centroEducativo.xsd.

Modifica el fichero act14_1.xml para que realice la validación descrita. **No pueden modificarse los esquemas XSD.**

14.2 Validación con dos esquemas no calificados

Vamos a realizar la misma validación del apartado anterior, pero con los elementos no calificados.

Modifica el atributo “elementFormDefault” de los dos esquemas (datosPersonales.xsd y centroEducativo.xsd) al valor “unqualified”.

Modifica el documento XML para que siga siendo válido.

(Recuerda que con elementos no calificados, lo únicos que deben llevar prefijo son los elementos globales).

Actividad 15. Vincular XML a XSD con varios espacios de nombres y sin espacios de nombres

Descarga el fichero act15.zip.

Abre el fichero act15.xml. Observa que tiene dos prefijos distintos en los elementos xml. Observa también que hay elementos sin prefijo.

Lo que se pretende es que la validación de este documento se realice empleando **tres** esquemas distintos:

- Los datos relativos al alumno tienen prefijo “dp” (datos personales) y se validarán con el esquema datosPersonales.xsd.
- Los datos relativos al centro educativo tienen prefijo “ce” (centro educativo) y se validarán con el esquema centroEducativo.xsd.
- Los datos relativos a la experiencia laboral se incluyen dentro del elemento <experiencia>, que no tiene prefijo. Estos datos se validarán con el esquema experienciaLaboral.xsd.

Modifica el fichero act15.xml para que realice la validación descrita. **No pueden modificarse los esquemas XSD.**

Actividad 16. Definir un elemento simple

a) Define un esquema XSD con un único elemento simple llamado “altura” cuyo contenido sea un número decimal.

   Crea un documento XML que se valide con ese esquema y sea válido.

b) Prueba a dejar el elemento “altura” vacío en el xml (sin contenido). ¿Es válido el documento?
   Añade el valor 180.35 como valor por defecto a la definición del elemento en el esquema. ¿Es válido ahora el documento?

c) Elimina el valor por defecto y añade 180.35 como valor fijo. Comprueba si el documento es válido:

   - Si se deja el contenido vacío
   - Si el contenido es un valor distinto a 180.35

Actividad 17. Definir elementos complejos con indicador de orden

17.1 Definir un único elemento complejo

Se quieren utilizar ficheros XML para que un programa registre datos de usuarios de una aplicación.

a) Define un esquema XSD con un elemento global denominado “usuario”. Dentro de ese elemento habrá cuatro elementos simples:

- “login”, de tipo texto
- “mail” de tipo texto
- “fecha_creación” de tipo fecha
- “id” de tipo número entero positivo

Se deberán introducir obligatoriamente los cuatro datos del usuario, pero pueden estar en cualquier orden.

Crea un fichero xml con los siguientes datos de un usuario:

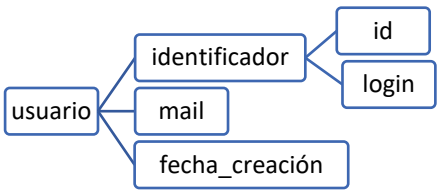
Id	125
Login	agarcia
Mail	ana.garcia@gmail.com
Fecha de creación	1 de enero de 2021

b) Modifica el apartado anterior para que obligatoriamente el orden de los datos sea el mostrado (login, mail, fecha_creación e id)

17.2 Anidar elementos complejos

Define un esquema XSD con un elemento global denominado “usuario”. Dentro de este elemento, estarán, en orden estricto, los siguientes elementos:

- “identificador”. Dentro de este elemento habrá dos elementos “id” y “login”. Solo es necesario que aparezca uno de los dos.
- “mail” de tipo texto
- “fecha_creación” de tipo fecha

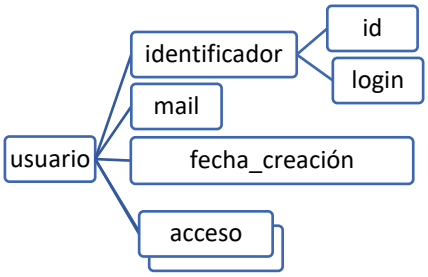


Introduce los datos del apartado anterior, eligiendo el “id” o el “login” (uno de los dos).

Actividad 18. Definir elementos complejos con indicador de ocurrencia

Modifica el ejercicio 17.2 para añadir un elemento:

- “acceso” de tipo fecha y hora: almacena la fecha y hora en la que el usuario ha accedido al sistema. Si nunca ha accedido, este elemento no aparecerá. Si ha accedido varias veces, habrá un elemento xml para cada acceso.



Crea un fichero XML válido e introduce los siguientes datos:

Id	125
Mail	ana.garcia@gmail.com
Fecha de creación	1 de enero de 2021
Acceso	1 de enero de 2021 17:00 2 de enero de 2021 10:30

Actividad 19. Elemento any

19.1 Validación con <any>

Define un esquema XSD de nombre “afición.xsd”. Este esquema definirá **tres elementos globales**:

- “deporte”: elemento simple de tipo texto
- “instrumento”: elemento simple de tipo texto
- “afición”: elemento compuesto en el que se podrán introducir de 1 a 5 elementos (cualquier elemento).

Comprueba que el siguiente xml es válido:

```
<afición xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:noNamespaceSchemaLocation="afición.xsd">
    <deporte>Balonmano</deporte>
    <instrumento>Violín</instrumento>
    <deporte>Tiro con arco</deporte>
</afición>
```

19.2 Elementos de varios esquemas

Añade al xml anterior las siguientes líneas justo antes de </afición>:

```
<o:otro>
  <categoría>Idioma</categoría>
  <descripción>Alemán</descripción>
</o:otro>
```

Ese nuevo elemento “otro” debe validarse con un nuevo esquema llamado “otro.xsd”, que debes crear. El resto de elementos (afición, deporte, instrumento) deben seguir validándose con el esquema “afición.xsd”, que no debes modificar.

Actividad 20. Insertar atributos

Para cargar de manera automática los datos de alumnos en un centro educativo, se emplea una aplicación que recibe los datos de las matriculaciones de cada curso en un fichero XML.

Descarga el archivo curso.zip, en el que hay un esquema y un ejemplo de datos XML.

Modifica el esquema XSD para añadir la siguiente información:

Información a incorporar	Dónde y cómo incorporarla	Observaciones
Grupo del curso	atributo del elemento curso	Obligatorio Valores de tipo texto
Nombre completo del alumno	atributo del elemento alumno	Obligatorio Valores de tipo texto
Alumno repetidor o no repetidor	atributo del elemento alumno	No obligatorio Posibles valores: true o false Por defecto tendrá el valor 'false'

Una vez modificado el esquema, comprueba que el esquema es válido.

Después incorpora los siguientes datos al fichero XML y comprueba que es válido:

- Este curso está referido al grupo 3ESO-F.
- La alumna con NIA 1256003 se llama Minerva Castell Ruíz.
- La alumna con NIA 1365220 se llama Lisa Sobrino Tejo y es repetidora.

Actividad 21. Elementos vacíos con atributos

Modifica la solución de la actividad 20 para que el elemento <alumno> sea un elemento vacío (sin contenido). Todos los datos de los alumnos aparecerán como atributos de ese elemento.

Se debe modificar el esquema XSD y el documento XML con los datos.

Actividad 22. Utilización de anyAttribute

22.1 Atributos en el mismo esquema

En el ejemplo de los alumnos de un grupo, vamos a añadir la posibilidad de insertar información adicional a cada alumno, a través de atributos.

Realiza los siguientes pasos:

- Modifica la solución de la actividad 21 para que el elemento <alumno>, además de los atributos que ya tiene asignados, admita cualquier otro atributo.
- En el esquema XSD, añade un atributo global:

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

- En el documento XML, añade la observación “Traslado desde otro centro educativo” a la alumna Lisa Sobrino Tejo.

22.2 Atributos en otro esquema

Siguiendo con el mismo ejemplo, vamos añadir atributos nuevos a los alumnos que se validarán con otro esquema llamado optativas.xsd, que debes descargar.

Modifica el fichero XML para poder añadir a los alumnos la siguiente información, como atributos, y que se valide con el esquema optativas.xsd.

Para ello debes asignar un espacio de nombres y un prefijo asociado al esquema optaivas.xsd.

Alumno	Optativas y extraescolares
NIA 1256003 (Minerva Castell Ruíz)	optativaCiencias: cálculo diferencial optativaArte: dibujo
NIA 1365220 (Lisa Sobrino Tejo)	Extraescolar: balonmano

Actividad 23. Restricciones en los tipos de datos

Crea un esquema llamado restricciones.xsd en que haya un objetos global: un elemento “teléfono”. Este elemento contendrá:

- Dos elementos: “tipo” y “número”, que aparecerán siempre y en orden.
  - El elemento “tipo” tendrá contenido de tipo texto y sólo admitirá los valores “móvil” y “fijo”.
  - El elemento “número” será de tipo numérico entero positivo y solo admitirá valores con 9 dígitos.

- Un atributo “nombre” tendrá contenido de tipo texto y solo admitirá textos de 10 a 20 caracteres.

Puedes crear un archivo xml que valide con este esquema para comprobar las restricciones.

Actividad 24. Tipos de datos personalizados

24.1 Tipos de datos personalizados con restricciones (simpleType)

Descarga el fichero act24_1.zip. Modifica el fichero act24_1.xsd para que las restricciones incluidas definan tipos de datos personalizados.

Por ejemplo el siguiente elemento:

```
<xs:element name="tipo">
<xs:simpleType>
     <xs:restriction base="xs:string">
            <xs:enumeration value="móvil"/>
            <xs:enumeration value="fijo"/>
     </xs:restriction>
</xs:simpleType>
</xs:element>
```

Pasa a ser:

```
<xs:element name="tipo" type="tipoTel"/>
```

Y se debe crear un tipo de datos (como global en el documento):

```
<xs:simpleType name="tipoTel">
    <xs:restriction base="xs:string">
         <xs:enumeration value="móvil"/>
         <xs:enumeration value="fijo"/>
     </xs:restriction>
</xs:simpleType>
```

Realiza la misma operación para todas las restricciones del documento, y comprueba que el fichero prueba.xml sigue siendo válido.

24.2 Tipos de datos personalizados con complexType

Se utilizan ficheros XML para enviar información sobre actividades municipales. Descarga el fichero act24_2.zip. Modifica el fichero act24_2.xsd para que el fichero xml sea válido.

Actividad 25. Tipos de datos con pattern

25.1 Patrones sencillos

Descarga el fichero pruebaPattern.zip. Modifica el patrón en cada apartado para añadir la restricción que se pide. Puedes comprobar su funcionamiento con el fichero prueba.xml.

a) Se admitan sólo 2 letras minúsculas

b) Se admitan sólo 2 letras mayúsculas o minúsculas

c) Se admita sólo una letra entre la a y la e, seguida de un espacio y un número

d) Sean dos palabras minúsculas de longitud indefinida cada una

e) Sea una palabra que obligatoriamente empiece por mayúscula

g) Una palabra que tenga el número 5 en su penúltima posición

h) Dos números de 5 cifras separados por un guion

25.2 Patrones aplicados

Descarga el fichero pruebaPattern2.zip. Modifica el patrón en cada apartado para añadir la restricción que se pide. Puedes comprobar su funcionamiento con el fichero prueba2.xml.

a) Define el patrón para el tipo de datos "**tipoFruta**" de forma que sólo admita tres posibles palabras: uva, manzana o pera (una de las tres). Sólo admite minúsculas.

b) Define el patrón para el tipo de datos "**tipoFruta2**" de forma que sólo admita tres posibles palabras: uva, manzana o pera (una de las tres). Deben estar en minúscula, pero se admite que la primera letra sea mayúscula (es válido uva y también Uva)

c) Código postal de Madrid. Debe tener 5 dígitos y empezar por 28. Define ese patrón para el tipo de datos "**cpMadrid**".

d) Código de seguridad. Debe tener 3 dígitos, un guion y 6 dígitos como mínimo. Ejemplo válido: 242-123421. Define ese patrón para el tipo de datos "**cs**".

e) Se va a utilizar una expresión regular para validar el nombre de una ciudad. Define el patrón para el tipo de datos "**nombreCiudad**". Debe cumplirse:

- El nombre de la cuidad debe empezar por mayúsculas
- No debe incorporar números ni caracteres especiales ($%& etc)
- Puede ser una ciudad compuesta por varias palabras separadas por un espacio. En este caso cada palabra debe empezar también por mayúscula.
- Puede estar formada por varias palabras separadas por un guion. En este caso cada palabra debe empezar también por mayúscula.

  Ejemplo: debe admitir las ciudades: Madrid, Ciudad Real, Rivas-Vaciamadrid, Patones De Arriba
  Ejemplo: no debe admitir: Ciudad real, Albastro2, Rivas-vaciamadrid