

# Actividades XML Schema

## 1. Cantidades limitadas

Crea un esquema que permita verificar un documento como el siguiente:

```
<precio>20</precio>
```

El elemento precio sólo podrá tener valores entre 3 y 99.

## 2. Cantidades limitadas con atributo divisa

Se desea crear un esquema para validar XML donde habrá un único elemento raíz llamado cantidad en el que debe ponerse siempre un atributo divisa que indique en qué moneda está una cierta cantidad. El atributo divisa siempre será una cadena y la cantidad siempre será un tipo numérico que acepte decimales (por ejemplo float o decimal).

El esquema debe validar los siguientes documentos:

```
<cantidad divisa="euro">20</cantidad >
```

```
<cantidad divisa="libra">15.32</cantidad >
```

Pero no debe validar ninguna de los siguientes:

```
<cantidad>20</cantidad>
```

```
<cantidad divisa="euro">daa</cantidad>
```

## 3. Cantidades limitadas con atributo divisa con sólo ciertos valores

Queremos ampliar la actividad anterior para evitar que errores como el siguiente:

```
<cantidad divisa="aaaa">18.32</cantidad>
```

Indicaremos que el atributo divisa sólo puede tener tres valores: euro, dólar y libra.

## 4. Códigos y sedes

Se necesita tener un esquema que valide un documento en el que sólo existe un elemento llamado código

- Dentro del elemento código hay una cadena con una estructura rígida: 2 letras mayúsculas, seguidas de 2 cifras, seguidas de 3 letras.
- El elemento código debe llevar un atributo suyo que será de tipo cadena.

## 5. Productos con atributos

Se desea crear un esquema que permita validar un elemento raíz llamado producto tipo xsd:string. El producto tiene dos atributos:

- Un atributo llamado cantidad obligatoria. Debe aceptar sólo números enteros positivos.
- Un atributo llamado unidad que sólo aceptará los valores xsd:string «cajas» y «palets».

## 6. Clientes con información adicional

Se desea crear un esquema XML que permita validar un elemento llamado cliente que pueda almacenar un xsd:string. El cliente contendrá:

- Un atributo obligatorio llamado código que contiene el código del cliente, que siempre consta de tres letras mayúsculas y tres números.
- Un atributo opcional llamado habitual que se utilizará para saber si es uno cliente habitual o no. Acepta valores true y false.
- Un atributo opcional llamado cantidad que indica la cantidad comprada. Debe ser un entero con valores entre 0 y 1000.

```
<cliente codigo="AAA222" habitual="true" cantidad="999">  
  Pepe Pérez </  
cliente>
```

## 7. Lista de códigos

Crea un esquema que permita validar un archivo como el siguiente:

```
<listacódigos>  
  <código>AAA2DD</código>  
  <código>BBB2EE</código>  
  <código>BBB2EE</código> </  
listacodis>
```

En concreto, todo código tiene la siguiente estructura:

- Primero van tres letras mayúsculas
- Después va un dígito.
- Por último, dos mayúsculas.

## 8. Lista de clientes

Crea un esquema que permita validar un archivo como el siguiente, en el que existe una lista de clientes y el nombre es opcional, pero los apellidos son obligatorios:

```
<listaclientes> <cliente>
  <nom>Juan</
    nom> <apellidos>Sanchez</
    apellidos> </client> <cliente>
<nombre>Jose</
nom>
  <apellidos>Diaz</
apellidos> </client> < /listaclientes>
```

## 9. Lista de alumnos

Crea un esquema para validar un listado de alumnos con las siguientes restricciones:

- La raíz es listadoalumnos. • Dentro

de este elemento existe uno o más elementos alumno. Todo alumno tiene siempre un atributo DNI que es obligatorio y que tiene una estructura formada por 7 o 8 cifras seguidas de una letra mayúscula. • Todo alumno tiene un elemento nombre y un ap1 obligatorios. • Todo alumno puede tener después del ap1 un elemento ap2 y un atributo edad, ambos son opcionales. • El elemento edad debe ser un número entero y positivo.

Un ejemplo de archivo:

```
<listado alumnos>
  <!--DNI atributo obligatorio--> <alumno dni="5667545Z">
  <!--Nombre y ap1 obligatorios-->
    <nombre>Jose</nom> <ap1>Sanchez</ap1> </
    alumno> < alumno
    dni="5778221D">

    <nom>Andres</nom> <ap1>Ruiz</
    ap1> <!--ap2 y edad son
    optativos--> <ap2>Ruiz</ap2> <!--La edad debe ser
    positivo--> <edad >25</
    edad> </alumno> </listado alumnos>
```