

EJERCICIO XSD

PÉREZ SELLERS, ALEJANDRO
LENGUAJE DE MARCAS 1º DAW

1. - Crea el XSD sin restricciones.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3    <xs:element name="producto">
4      <xs:complexType>
5        <xs:sequence>
6          <xs:element name="nombre" type="xs:string"/>
7          <xs:element name="precio" type="xs:decimal"/>
8        </xs:sequence>
9        <xs:attribute name="id" type="xs:string"/>
10     </xs:complexType>
11   </xs:element>
12 </xs:schema>
13
```

Contiene dos elementos llamados nombre de tipo xs:string (texto) y precio de tipo xs:decimal (número decimal). Además, posee un atributo id asociado al elemento producto que es de tipo xs:string

- 2.- Crea las siguientes restricciones: Que el precio no pueda ser negativo. (Se puede hacer definiéndolo de un tipo determinado o con restricciones) Que el id cumpla el patrón indicado, con mayúsculas y minúsculas. ¿Un atributo puede tener restricciones? El precio tiene que tener dos decimales.

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
3    <xs:element name="producto">
4      <xs:complexType>
5        <xs:sequence>
6          <xs:element name="nombre" type="xs:string"/>
7          <xs:element name="precio">
8            <xs:simpleType>
9              <xs:restriction base="xs:decimal">
10               <xs:minInclusive value="0.00"/>
11               <xs:fractionDigits value="2"/>
12             </xs:restriction>
13           </xs:simpleType>
14         </xs:element>
15       </xs:sequence>
16       <xs:attribute name="id">
17         <xs:simpleType>
18           <xs:restriction base="xs:string">
19             <xs:pattern value="[A-Za-z0-9] +"/>
20           </xs:restriction>
21         </xs:simpleType>
22       </xs:attribute>
23     </xs:complexType>
24   </xs:element>
25 </xs:schema>
26

```

Se define un elemento producto que esta formado por nombre de tipo xs:string texto, precio un elemento de tipo xs:decimal que tiene un valor mínimo de 0.00 y un limite de 2 decimale fractionDigits.

El id atributo tiene una validación basada en un patrón que permite caracteres alfanuméricos [A-Za-z0-9] +)

```

<curso codigo="DAM101">
  <nombre>Lenguaje de Marcas</nombre>
  <profesor nombre="Fabián" email="fabian@ejemplo.com"/>
  <duracion horas="60"/>
</curso>

```

1.- Crea el XSD sin restricciones

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="curso">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="profesor">
          <xs:complexType>
            <xs:attribute name="nombre" type="xs:string" use="required"/>
            <xs:attribute name="email" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
        <xs:element name="duracion">
          <xs:complexType>
            <xs:attribute name="horas" type="xs:string" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="codigo" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

2.- Crea las siguientes restricciones:

- El código tiene que ser del estilo de DAM101, es decir, 6 caracteres, 3 letras y 3 números.
- Hay que validar que el email sea correcto.
- Las horas tienen que estar entre 60 y 120

Piensa que los atributos tienen que ser obligatorios

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="curso">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nombre" type="xs:string"/>
        <xs:element name="profesor">
          <xs:complexType>
            <xs:attribute name="nombre" type="xs:string"
use="required"/>

```

```

        <xs:attribute name="email" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:string">
                    <xs:pattern value="^[a-zA-Z0-9._%+-
]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
<xs:element name="duracion">
    <xs:complexType>
        <xs:attribute name="horas" use="required">
            <xs:simpleType>
                <xs:restriction base="xs:integer">
                    <xs:minInclusive value="60"/>
                    <xs:maxInclusive value="120"/>
                </xs:restriction>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="codigo" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="[A-Z]{3}[0-9]{3}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:schema>

```

Los atributos obligatorios en este código son nombre y email en el elemento profesor, horas en el elemento duración y código en el elemento principal esta marcados como use="required".

Se validan elementos como email, utilizando un patrón `^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$` que valida direcciones de correo electrónico y el atributo código tiene un patrón `[A-Z]{3}[0-9]{3}` que exige tres letras mayúsculas seguidas de tres dígitos.

Las restricciones del atributo horas con `xs:minInclusive` y `xs:maxInclusive` que limitan su valor entre 60 y 120.

```
<vehiculo matricula="1234ABC" tipo="turismo">
  <marca>Renault</marca>
  <modelo>Clio</modelo>
  <anio>2020</anio>
</vehiculo>
```

Esto es un elemento que está dentro de un elemento principal que se llama concesionario que tiene varios vehículos.

Crea un XSD que:

- Valide la matrícula
- Tipo solo puede ser: turismo, comercial, moto y camión
- Solo hay vehículos a partir del año 1990

RECUERDA QUE LOS ATRIBUTOS NO PUEDEN ESTAR DENTRO DE UN SEQUENCE Y QUE EL SIMPLETYPE PUEDE ESTAR ANIDADO DENTRO O SACARLO FUERA SI LE PONES NOMBRE AL TIPO

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="concesionario">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="vehiculo" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="marca" type="xs:string"/>
              <xs:element name="modelo" type="xs:string"/>
              <xs:element name="anio">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minInclusive value="1990"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute name="matricula" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:pattern value="[0-9]{4}[A-Z]{3}"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="tipo" use="required">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="turismo"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```
        <xs:enumeration value="comercial"/>
        <xs:enumeration value="moto"/>
        <xs:enumeration value="camión"/>
    </xs:restriction>
</xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Para validar la matricula, se usa un atributo matricula con una restricción xs:pattern para validar el formato.

Se uso un atributo tipo, se usa un xs:enumeration con los valores posibles turismo, comercial, moto, camion.

El elememento anio (año) se escribe de esa manera para que no de fallo, tiene una restricción xs:minInclusive para asegurar que solo se acepten años a partir de 1990.

Algunos de los simpleType están anidados dentro de los elementos o atributos, según indicaciones.

```

<pedido numero="1001" fecha="2025-03-24">
  <producto id="P01" cantidad="2">
    <nombre>Ratón</nombre>
    <precio>15.50</precio>
  </producto>
  <producto id="P02" cantidad="1">
    <nombre>Teclado</nombre>
    <precio>25.00</precio>
  </producto>
</pedido>

```

Esto es un elemento que está dentro de un elemento principal que se llama "orders"

Crea un XSD que:

- Valide la fecha para que sea ese formato
- Los atributos tienen que ser obligatorios.
- El número de pedido y la cantidad tienen que tener el formato adecuado (selecciónalo tú siguiendo criterios lógicos)
- El id de producto tiene que seguir un patrón.
- El precio no puede superar las 5 cifras.
- RECUERDA QUE LOS ATRIBUTOS NO PUEDEN ESTAR DENTRO DE UN SEQUENCE

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="orders">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="pedido" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="producto" maxOccurs="unbounded">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="nombre" type="xs:string"/>
                    <xs:element name="precio">
                      <xs:simpleType>
                        <xs:restriction base="xs:decimal">
                          <xs:totalDigits value="5"/>
                        </xs:restriction>
                      </xs:simpleType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:attribute name="id" use="required">

```



```

        <xs:simpleType>
            <xs:restriction base="xs:string">
                <xs:pattern value="P\d{4}"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="cantidad" use="required">
        <xs:simpleType>
            <xs:restriction base="xs:integer">
                <xs:minInclusive value="1"/>
            </xs:restriction>
        </xs:simpleType>
    </xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="numero" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{4}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
<xs:attribute name="fecha" use="required">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern value="\d{4}-\d{2}-\d{2}"/>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

El atributo fecha usa un patrón \d{4}-\d{2}-\d{2} para verificar el formato YYYY-MM-DD. Además, los atributos número, fecha, id y cantidad están marcados como use="required".

El formato adecuado para número, tiene un patrón \d{4}, por ejemplo un número de 4 dígitos y cantidad mayor o igual a 1. El patrón para id de producto se valida con el patrón P\d{4}, es una letra p seguida de 4 dígitos.

Precio con máximo de 5 cifras se usa xs:decimal con una restricción de xs:totalDigits.

