

EJERCICIO 1 (ej1.xml, ej1.dtd):

Crea un documento DTD externo asociado al siguiente documento XML de forma que sea válido. Se deben tener en cuenta las siguientes características:

- El número de artículo (n_art) es un valor único por artículo y es obligatorio.
- La cantidad del artículo puede no aparecer.
- La unidad de medida puede no aparecer, pero si lo hace debe tomar el valor "Sistema Internacional".
- Un artículo puede no tener ofertas o disponer de varias.
- El tipo de marca puede no aparecer, y tomar los valores "nacional" o "internacional".

```
<!ELEMENT supermercado (articulo+)>
<!ELEMENT articulo (descripcion, marca, medida, seccion, oferta*,
perecedero?)>
  <!ATTLIST articulo n_art ID #REQUIRED>
  <!ATTLIST articulo cantidad CDATA #IMPLIED>
<!ELEMENT descripcion (#PCDATA)>
<!ELEMENT marca (#PCDATA)>
  <!ATTLIST marca tipo (nacional|internacional) #IMPLIED>
<!ELEMENT medida (#PCDATA)>
  <!ATTLIST medida unidad CDATA #FIXED "Sistema Internacional">
<!ELEMENT seccion (#PCDATA)>
<!ELEMENT oferta (#PCDATA)>
<!ELEMENT perecedero (caducidad)>
<!ELEMENT caducidad (#PCDATA)>
```

Ejercicio 2 (ej2.xml, ej2.dtd)

A partir del siguiente DTD, elabora un XML con al menos tres elementos con datos ficticios para el que sea válido:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libreria SYSTEM "ej2.dtd">
<libreria catalogo="123455">
  </libro titulo="Nada" editorial="Planeta" autor="Carmen Laforet">
  <revista marca="1234">
    <numero>23</numero>
    <titular>National Geographic</titular>
    <editorial>NewWorld</editorial>
```

```

</revista>
<revista marca="5678">
  <numero>14</numero>
  <titular>Nuevos avances</titular>
  <editorial>La Editora Moderna</editorial>
</revista>

</periodico fecha="23-12-2020" nombre="El Periódico">
</periodico fecha="15-04-2012" nombre="New York Times"
version_web="www.nyt.com">
</libreria>

```

EJERCICIO 3 (ej3.xml, ej3.xsd):

Crea un XSD asociado al siguiente documento XML de forma que sea válido, cumpliendo los siguientes requerimientos:

- El elemento teléfono será un entero y dicho elemento podrá aparecer entre 0 y 5 veces.
- El elemento dirección podrá aparecer 1 o 2 veces.
- El elemento número será un entero entre 0 y 200.
- El elemento provincia sólo podrá contener Sevilla o Huelva.
- El atributo dni es obligatorio, y será una plantilla de 8 números y una letra mayúscula.
- El elemento edad será opcional.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="alumnos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="alumno" maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre" type="xs:string"/>
              <xs:element name="direccion" maxOccurs="2">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="calle"
type="xs:string"/>
                    <xs:element name="numero">

```

```

<xs:simpleType>
  <xs:restriction
base="xs:integer">
    <xs:minInclusive
value="0"/>
    <xs:maxInclusive
value="200"/>
  </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="ciudad"
type="xs:string"/>
<xs:element name="provincia">
  <xs:simpleType>
    <xs:restriction
base="xs:string">
      <xs:enumeration
value="Sevilla" />
      <xs:enumeration
value="Huelva" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="telefono"
type="xs:integer" minOccurs="0" maxOccurs="5">
  </xs:element>
<xs:element name="edad" type="xs:integer"
minOccurs="0"/>
</xs:sequence>
<xs:attribute name="dni">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:pattern
value="\d{8}[A-Z]"></xs:pattern>
    </xs:restriction>
  </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>

```

```

    </xs:complexType>
  </xs:element>
</xs:schema>

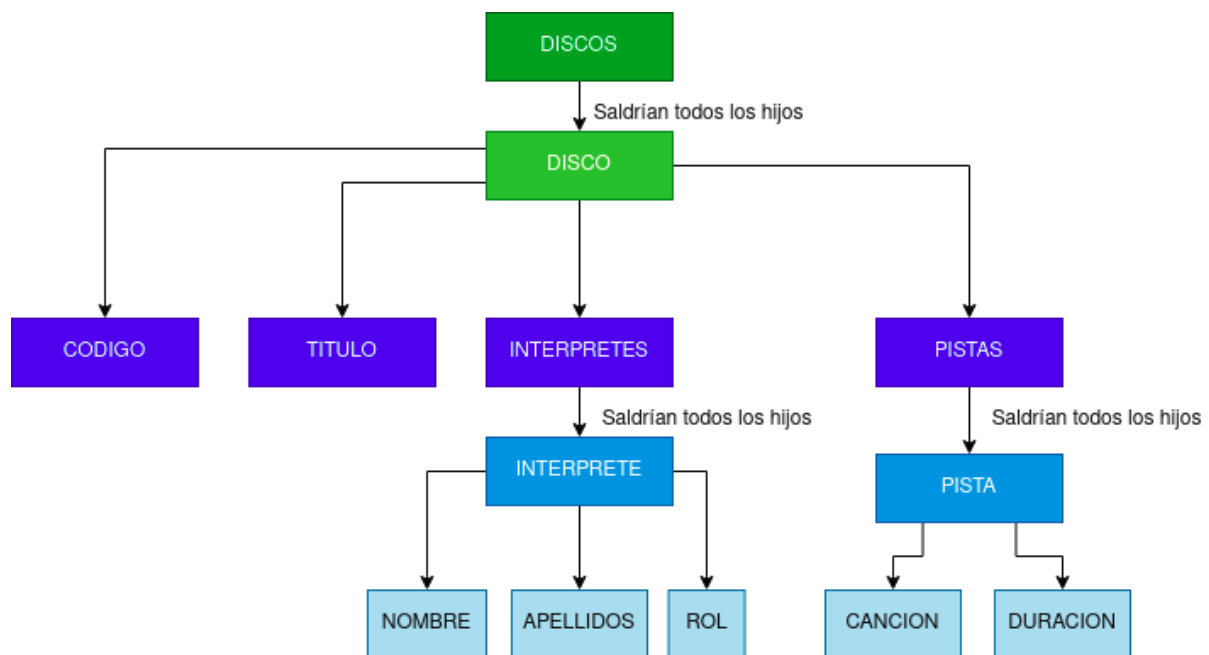
```

EJERCICIO 4 (ej4.pdf ó ej4.jpg, ej4.xsd):

Dibuja un diagrama de árbol (ej4.pdf ó ej4.jpg) que recoja la siguiente información:

Una discográfica quiere tener una base de datos de su colección de discos.

- Para cada disco, quiere guardar su código, título, el/los intérpretes, y el número de pistas.
 - El código constará de una letra mayúscula que puede ser R (rock), P (pop), C (clásica) u O (otra), seguida de 3 dígitos. (Ejemplo: P032)
 - Para cada intérprete, quiere guardar el nombre y apellidos (obligatorio), así como su rol o función si se conoce (ej. solista, guitarrista, ...)
 - Para cada pista, se quiere almacenar el nombre de la canción y su duración.
 - Cada elemento/atributo debe definirse con el tipo más adecuado a los datos que contiene.
- Genera un archivo ej4.xsd que recoja la estructura con los requisitos anteriores.



```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="discos">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="disco" maxOccurs="unbounded">

```

```

        <xs:complexType>
            <xs:sequence>
                <xs:element name="codigo">
                    <xs:simpleType>
                        <xs:restriction base="xs:string">
                            <xs:pattern
value="[RPCO]\d{3}"></xs:pattern>
                        </xs:restriction>
                    </xs:simpleType>
                </xs:element>
                <xs:element name="titulo" type="xs:string"
/>
            <xs:element name="interpretes">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="interprete"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element
name="nombre" type="xs:string"/>
                                    <xs:element
name="apellido" type="xs:string"/>
                                    <xs:element
name="rol" type="xs:string" minOccurs="0"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="pistas">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="pista"
maxOccurs="unbounded">
                            <xs:complexType>
                                <xs:sequence>
                                    <xs:element
name="cancion" type="xs:string"/>
                                    <xs:element
name="duracion" type="xs:string"/>
                                </xs:sequence>
                            </xs:complexType>
                        </xs:element>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

```
        </xs:complexType>
    </xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="dni">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:pattern
value="\d{8}[A-Z]"></xs:pattern>
        </xs:restriction>
    </xs:simpleType>
</xs:attribute>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```