EJERCICIO 1

Realizar marcadores.xsd que valide el siguiente documento xml

```
<?xml version="1.0" encoding="UTF-8"?>
<marcadores
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="marcadores.xsd">
   <pagina>
      <nombre>Abrirllave</nombre>
      <descripcion>Tutoriales de
informática.</descripcion>
      <url>http://www.abrirllave.com/</url>
   </pagina>
   <pagina>
      <nombre>Wikipedia</nombre>
      <descripcion>La enciclopedia
libre.</descripcion>
      <url>http://www.wikipedia.org/</url>
   </pagina>
   <pagina>
      <nombre>W3C</nombre>
      <descripcion>World Wide Web
Consortium.</descripcion>
      <url>http://www.w3.org/</url>
   </pagina>
</marcadores>
```

EJERCICIO 2

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
   <ficha numero="01" letra="Z">
      <codigo>11</codigo>
      <nombre>Ana Sanz Tin</nombre>
      <edad>22</edad>
   </ficha>
   <ficha numero="02">
      <codigo>12</codigo>
      <nombre>Iker Rubio Mol</nombre>
      <edad>23</edad>
   </ficha>
</fichas>
```

Escribir el contenido del archivo ***"fichas.xsd"*** que permita validarlo, teniendo en cuenta que:

- se debe definir la "edad" con la restricción de que el valor que tome no pueda ser menor que 0 ni mayor que 130
- Tanto el atributo **numero** como el elemento "código" utilizan la misma restricción que solamente les permite tomar un valor entero expresado con dos dígitos comprendidos entre "**00**" y "**19**".
- El atributo **letra puede** tomar por valor una de las siguientes letras: "**X**", "**Y**" o "**Z**". La restricción debe definirse de forma que solamente pueda ser utilizada por dicho atributo.
- Para cada ficha se tiene que indicar un número, obligatoriamente. Sin embargo, la letra es opcional.

EJERCICIO 3

```
<?xml version="1.0" encoding="UTF-8"?>
<articulos
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="articulos.xsd">
   <articulo>
      <nombre>Mesa</nombre>
      <precio moneda="Euro">50</precio>
   </articulo>
   <articulo>
      <nombre>Silla</nombre>
      <precio moneda="Dólar">78.99</precio>
   </articulo>
</articulos>
```

Realizar el documento artículos.xsd teniendo en cuenta:

- el elemento "precio" puede tomar por valor un número que contenga tres dígitos como máximo y, de ellos, solamente dos pueden ser decimales.
- La moneda puede tomar los valores euro y dólar

EJERCICIO 4

Dado el siguiente schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="fichas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ficha"
maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre"
type="xs:string"/>
              <xs:element name="iniciales">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[A-Z][A-Z][A-
Z]"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="edad"
type="xs:integer"/>
             </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Corregir los errores del siguiente documento XML para que sea válido:

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
   <ficha>
      <nombre>Antonio Machado Ruiz</nombre>
      <iniciales>AMR</iniciales>
      <edad>22</edad>
   </ficha>
   <ficha>
      <nombre>Mario Moreno</nombre>
      <iniciales>MM</iniciales>
      <edad>23</edad>
   </ficha>
   <ficha>
      <iniciales>ALO</iniciales>
      <nombre>Ada Lovelace</nombre>
      <edad>24</edad>
   </ficha>
   <ficha>
      <nombre>pablo ruiz picasso</nombre>
      <iniciales>prp</iniciales>
      <edad>24</edad>
   </ficha>
</fichas>
```

EJERCICIO 5

Dado el siguiente schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="fichas">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="ficha"
maxOccurs="unbounded">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="nombre"
type="xs:string"/>
              <xs:element name="iniciales">
                <xs:simpleType>
                  <xs:restriction base="xs:string">
                    <xs:pattern value="[A-Z][A-Z][A-
Z]"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
              <xs:element name="edad"
type="xs:integer"/>
             </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Realizar los cambios necesarios en el schema para que el siguiente xml sea válido

```
<?xml version="1.0" encoding="UTF-8"?>
<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
   <ficha>
      <nombre>Ana Sanz Tin</nombre>
      <iniciales>AST</iniciales>
      <edad>22</edad>
      <iniciales-al-reves>TSA</iniciales-al-reves>
   </ficha>
   <ficha>
      <nombre>Iker Rubio Mol</nombre>
      <iniciales>IRM</iniciales>
      <edad>23</edad>
      <iniciales-al-reves>MRI</iniciales-al-reves>
   </ficha>
</fichas>
```

EJERCICIO 6

```
<?xml version="1.0" encoding="UTF-8"?>
<ubicaciones
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="ubicaciones.xsd">
   <ubicacion metros="32">norte</ubicacion>
   <ubicacion metros="25">este</ubicacion>
   <ubicacion metros="64">este</ubicacion>
</ubicaciones>
```

Realizar el documento ubicaciones.xsd teniendo en cuenta:

- La ubicación puede tomar los valores: norte, sur, este y oeste.
- Los metros son enteros positivos.

EJERCICIO 7

```
<?xml version="1.0" encoding="UTF-8"?>
<muebles xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xsi:noNamespaceSchemaLocation="muebles.xsd">
   <mueble color="blanco">mesa</mueble>
   <mueble color="gris">silla</mueble>
</muebles>
```

Realizar el documento muebles.xsd teniendo en cuenta:

- El color del mueble puede tomar los valores: blanco, gris, negro y wenge.
- El mueble puede tomar los valores: armario, mesa y silla.

EJERCICIO 8

Dado el archivo bingo.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:element name="bingo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="bola" maxOccurs="unbounded">
          <xs:complexType>
            <xs:attribute name="numero"
type="numeroDeBola"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name="numeroDeBola">
     <xs:restriction base="xs:positiveInteger">
        <xs:minInclusive value="1"/>
        <xs:maxExclusive value="90"/>
     </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

Escribir el código de un documento XML que pueda ser validado por ***"bingo.xsd"*** y almacene los números 17, 23 y 65.

EJERCICIO 9

Dado el archivo personas.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="personas" type="tpersonas" />

<xs:complexType name="tpersonas" >
   <xs:sequence>
    <xs:element ref="persona" maxOccurs="unbounded">
      </xs:sequence>
    </xs:complexType>

<xs:element name="persona">
 <xs:complexType mixed="true">
    <xs:sequence>
      <xs:element name="nombre" type="xs:string"/>
      <xs:element name="ciudad" type="xs:string"/>
      <xs:element name="edad"
type="xs:positiveInteger"/>
     </xs:sequence>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Utilizando los elementos "nombre", "ciudad" y "edad", escribir el código de un documento XML que pueda ser validado por ***"personas.xsd"*** y que almacene la siguiente información:

- "Eva vive en París y tiene 25 años."
- "Giovanni vive en Florencia y tiene 26 años."

EJERCICIO 10

Construir el archivo xsd que valide el siguiente documento XML.

- Los tipos de datos han de ser los adecuados al tipo de información a almacenar
- No se permite el uso del atributo ref
- Debe incluir la posibilidad de que existiese un número ilimitado de elementos nota.
- Los atributos día y hora son opcionales y num es obligatorio
- Los elementos para, de, titulo y contenido pueden aparecer en cualquier orden.

```
<lista_de_notas>
<nota num="11" dia="2011-02-02" hora="10:10:10">
<para>José</para>
<de>Ana</de>
<titulo>Cita</titulo>
<contenido>Nos vemos el sábado a las 15:00</contenido>
</nota>
</lista_de_notas>
```

EJERCICIO 11

Crea un DTD interno y otro externo asociado al siguiente documento XML de forma que sea válido. Comprobar la buena formación y la validez del documento en ambos casos. 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 y su valor no puede contener caracteres especiales.
- 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".

```
<?xml version="1.0" encoding="UTF-8"?>
<supermercado>
<articulo n_art="articulo1" cantidad="300">
<descripcion>Yogurt</descripcion>
<marca tipo="nacional">Danone</marca>
<medida unidad="Sistema Internacional">1 Pack 4</medida>
<seccion>Lacteos</seccion>
</articulo>
<articulo n_art="articulo2">
<descripcion>Queso fresco</descripcion>
<marca tipo="nacional">Burgo de Arias</marca>
<medida>250 gr</medida>
<seccion>Lacteos</seccion>
<oferta>Lote descuento</oferta>
<perecedero>
<fabricacion>Febrero de 2014</fabricacion>
</perecedero>
</articulo>
</supermercado>
```

EJERCICIO 12

Dado el siguiente DTD, construye un documento XML válido para el mismo:

```
<!ELEMENT Libro (Titulo, Seccion, SubSeccion?, Contenido, Copyright)>
<!ATTLIST Libro Catalogo ID #REQUIRED>
<!ELEMENT Titulo (#PCDATA)>
<!ELEMENT Seccion (#PCDATA)>
<!ELEMENT SubSeccion (#PCDATA)>
<!ELEMENT Contenido ((Capitulo+)|(Capitulo+, Separacion?)+)>
<!ELEMENT Capitulo (Tema+)>
<!ATTLIST Capitulo materia (XML|XHTML) "XML">
<!ELEMENT Tema (#PCDATA)>
<!ELEMENT Separacion EMPTY>
<!ELEMENT Copyright (#PCDATA)>
```

EJERCICIO 13

Dado el siguiente XSD construye un documento XML válido para el mismo:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="pedido">
<xs:complexType>
<xs:sequence>
<xs:element name="destino" type="TipoDireccion"/>
<xs:element name="ordenante" type="TipoDireccion"/>
<xs:element name="observaciones" type="xs:string" minOccurs="0"/>
<xs:element name="contenido">
<xs:complexType>
<xs:sequence>
<xs:element name="producto" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<xs:element name="nombre" type="xs:string"/>
<xs:element name="cantidad" type="xs:integer"/>
<xs:element name="precio" type="xs:decimal"/>
<xs:element name="observaciones" type="xs:string" minOccurs="0"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="fecha" type="xs:date" use="required"/>
</xs:complexType>
</xs:element>

<xs:complexType name="TipoDireccion">
<xs:sequence>
<xs:element name="nombre" type="xs:string"/>
<xs:element name="direccion" type="xs:string"/>
<xs:element name="ciudad" type="xs:string"/>
<xs:element name="codpostal">
<xs:simpleType>
<xs:restriction base="xs:positiveInteger">
<xs:minInclusive value="1000"/>
<xs:maxInclusive value="60000"/>
</xs:restriction>
</xs:simpleType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:schema>
```