

RELACIÓN 3 - SCHEMAS

1. Para los siguientes elementos:

```
<ciudad>Roma</ciudad>
```

```
<hora>18:29:45</hora>
```

```
<nota>7.5</nota>
```

```
<apto>true</apto>
```

Escribir sus definiciones simples correspondientes.

- **< ciudad >**: Un elemento simple que contiene el nombre de una ciudad.
- **< hora >**: Un elemento simple que contiene una hora en el formato "hora:minutos:segundos".
- **< nota >**: Un elemento simple que contiene una calificación en formato decimal.
- **< apto >**: Un elemento simple que indica si una persona está apta o no, representado por un valor booleano.

2. Definir un elemento llamado "puertaCerrada" de tipo lógico, que por defecto tenga el valor "falso", y otro elemento llamado "ventanaAbierta" también de tipo lógico, que tenga asignado el valor fijo "verdadero".

```
<?xml version="1.0"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="root">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="puertaCerrada" type="xs:boolean" default="false"/>

        <xs:element name="ventanaAbierta" type="xs:boolean" fixed="true"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Ejemplo:

```
<puertaCerrada>false</puertaCerrada>
<ventanaAbierta>true</ventanaAbierta>
```

3. Dado el siguiente documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>

<fichas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="fichas.xsd">
  <ficha numero="1">
    <nombre>Ana Sanz Tin</nombre>

    <edad>22</edad>
  </ficha>

  <ficha numero="2">
    <nombre>Iker Rubio Mol</nombre>

    <edad>23</edad>
  </ficha>
</fichas>
```

Escribir el contenido del archivo "fichas.xsd" que permita validarlo.

```
<?xml version="1.0"?>
<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="edad" type="xs:integer"/>
            </xs:sequence>

            <xs:attribute name="numero" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

4. Utilizando el mismo documento XML del ejercicio anterior, 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. Además, en vez de "xs:minInclusive" y "xs:maxInclusive", se debe utilizar:

- "xs:minExclusive": que sirve para especificar que el valor debe ser mayor que el indicado.
- "xs:maxExclusive": que sirve para especificar que el valor debe ser menor que el indicado.

Respuesta:

```

<?xml version="1.0"?>

<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="edad">
                <xs:simpleType>
                  <xs:restriction base="xs:integer">
                    <xs:minExclusive value="-1"/>

                    <xs:maxExclusive value="131"/>
                  </xs:restriction>
                </xs:simpleType>
              </xs:element>
            </xs:sequence>

            <xs:attribute name="numero" type="xs:integer" use="required"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

5. Determina y corrige el/los error/errores que provoca que el fichero XML no esté bien formado.

```

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE etiqueta[
  <!ELEMENT etiqueta (nombre, calle, ciudad, pais, codigo)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT calle (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
  <!ELEMENT pais (#PCDATA)>
  <!ELEMENT codigo (#PCDATA)>
]>

<nombre>Pepe García</nombre>
<calle>C/Ronda, 3</calle>
<pais>España<pais>
<codigo>18465</codigo>
</etiqueta>

```

Corrección:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE etiqueta[
  <!ELEMENT etiqueta (nombre, calle, ciudad, pais, codigo)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT calle (#PCDATA)>
  <!ELEMENT ciudad (#PCDATA)>
  <!ELEMENT pais (#PCDATA)>
  <!ELEMENT codigo (#PCDATA)>
]>

<etiqueta>
  <nombre>Pepe García</nombre>

  <calle>C/Ronda, 3</calle>

  <ciudad>Granada</ciudad>

  <pais>España</pais>

  <codigo>18465</codigo>
</etiqueta>
```

XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="etiqueta" type="etiquetaType"/>
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="calle" type="xs:string"/>
  <xs:element name="ciudad" type="xs:string"/>
  <xs:element name="pais" type="xs:string"/>
  <xs:element name="codigo" type="xs:string"/>

  <xs:complexType name="etiquetaType">
    <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:element ref="calle"/>
      <xs:element ref="ciudad"/>
      <xs:element ref="pais"/>
      <xs:element ref="codigo"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

6. Explica esta línea y realiza su equivalente en XML Schema (xsd):

```
<!ELEMENT receta (titulo, ingrediente+, procedimiento+, tiempo?)>
```

Explicación:

La línea "" es una declaración de tipo de elemento en DTD (Document Type Definition) que especifica la estructura de un elemento llamado "receta". Esta línea indica que un elemento "receta" debe tener como sub-elementos un elemento "titulo", uno o más elementos "ingrediente", uno o más elementos "procedimiento", y un elemento opcional "tiempo".

Equivalente:

```
<xs:element name="receta">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="titulo" type="xs:string"/>

      <xs:element name="ingrediente" type="xs:string" maxOccurs="unbounded"/>

      <xs:element name="procedimiento" type="xs:string" maxOccurs="unbounded"/>

      <xs:element name="tiempo" type="xs:string" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

7. Escribir un XML Schema para el siguiente documento XML, e incluir los cambios necesarios en el mismo para referenciar al esquema creado. Se sabe que todos los elementos hijos del elemento "nota" son obligatorios y deben aparecer solo una vez.

```
<?xml version="1.0" encoding="UTF-8"?>

<nota>
  <para>Pedro</para>

  <de>Laura</de>

  <titulo>Recordatorio</titulo>

  <contenido>A las 7:00 pm en la puerta del teatro</contenido>
</nota>
```

XML Schema:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="nota" type="notaType"/>

  <xs:complexType name="notaType">
    <xs:sequence>
      <xs:element name="para" type="xs:string"/>

      <xs:element name="de" type="xs:string"/>

      <xs:element name="titulo" type="xs:string"/>

      <xs:element name="contenido" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>

```

Cambios en el XML para referenciar al esquema:

```

<?xml version="1.0" encoding="UTF-8"?>

<nota xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="nombre_del_archivo.xsd">
  <para>Pedro</para>

  <de>Laura</de>

  <titulo>Recordatorio</titulo>

  <contenido>A las 7:00 pm en la puerta del teatro</contenido>
</nota>

```