XML. Ejercicios XML Schemas.

1. Ejercicio “RUTAS”

Se desea generar una estructura de archivo, en formato XML, para validar cada ruta de turismo (montaña, monumentos, etc.) del sistema.
(El presente ejercicio desarrolla, paso a paso, el Esquema XML para la validación de las rutas, conforme al documento XML que se indica).

► Se pide: generar el documento XML “ruta01.xml” y el Esquema XML “rutas.xsd”, siguiendo las instrucciones que se detallan a continuación. Comprobar que está bien formado y validar.

(Nota: texto con fondo verde se corresponde con el enunciado. Texto con fondo azul se corresponde con la solución. Se efectúa la resolución del ejercicio siguiendo el modelo de diseño de Tipos con Nombres Reutilizables).

▫ Documento XML: “ruta01.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
<ruta cod="R01" tiporuta="Montaña" dificultad=”Baja”>
   <denominacion>Ruta de las Cascadas</denominacion>
   <distancia>8 Km</distancia>
   <puntosinteres>
      <punto pos="P01">
         <nombre>Cascada 1</nombre>
         <puntokilometrico>1.5</puntokilometrico>
      </punto>
      <punto pos="P02">
         <nombre>Cascada 2</nombre>
         <puntokilometrico>2.3</puntokilometrico>
      </punto>
      <punto pos="P03">
         <nombre>Cascada 3</nombre>
         <coordenadas>259;367</coordenadas>
      </punto>
   </puntosinteres>
   <plano tipo="JPG">MapaR01.jpg</plano>
</ruta>
```

La primera línea del documento XML nos indica el tipo de documento, la versión utilizada y la codificación.

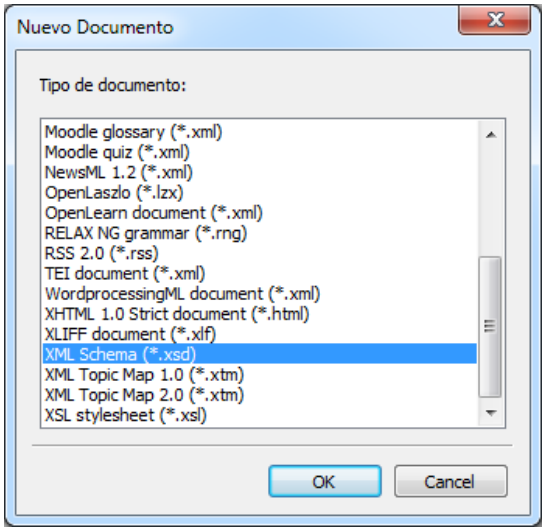
Como paso previo identificamos el nodo raíz en el documento XML, en este caso “ruta”.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruta cod="R01" … >
   ...
</ruta>
```

1.1. Crear archivo Esquema XML.

El primer paso será crear el documento con el Esquema XML, al que denominaremos "rutas.xsd".

▫ Esquema XML: "rutas.xsd"



Insertamos la cabecera del esquema. Declaramos el tipo de documento y el elemento raíz esquema (<xs:schema … > … </xs:schema>).

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

(Nota: indicamos el atributo "elementFormDefault=”qualified" para indicar que en la declaración de elementos del espacio de nombres "http://www.w3.org/2001/XMLSchema" se utilizará el prefijo establecido, en este caso "xs")

1.2. Asociar documento XML a Esquema XML.

A continuación, procedemos a insertar en el documento XML su asociación con el Esquema XML creado.

```
<ruta xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xsi:noNamespaceSchemaLocation="rutas.xsd"
… >
```

▫ Documento XML: "ruta01.xml"

De esta forma la cabecera del documento XML quedará de la siguiente manera:

```
<?xml version="1.0" encoding="UTF-8"?>
<ruta xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="rutas.xsd"
    cod="R01" tiporuta="Montaña" dificultad="Baja">
```

1.3. Declaración inicial de elementos en el Esquema XML.

A partir de este momento comenzaremos a desarrollar el Esquema XML, desgranando el árbol de nodos poco a poco, comenzando por el nodo raíz, "ruta".

```
<ruta … >
   <denominacion>…</denominacion>
   <distancia>…</distancia>
   <puntosinteres>
        <punto … >
             <nombre>…</nombre>
             <puntokilometrico>…</puntokilometrico>
        </punto>
        <punto … >
             <nombre>…</nombre>
             <coordenadas>…</coordenadas>
        </punto>
        …
   </puntosinteres>
   <plano … >…</plano>
</ruta>
```

Elementos identificados: por cada elemento definiremos cuál es su tipo, que posteriormente desarrollaremos. Fijaremos el tipo para aquellos elementos cuya información sea del tipo simple predefinido. Nombraremos un tipo específico para el resto (en principio lo nombramos y posteriormente lo desarrollaremos).

- "ruta": elemento raíz del documento XML. [Descendientes y atributos: Tipo Complejo, que denominaremos "tipoRuta"]
  `<xs:element name="ruta" type="tipoERuta"/>`

- "denominacion": contenido textual, sin descendientes ni atributos. [Sería del tipo simple predefinido, cadena, pero vamos a indicarle un tipo simple construido para agregarle restricciones: Tipo Simple Construido, que denominaremos "tipoCadena"]
  `<xs:element name="denominacion" type="tipoCadena"/>`

- "distancia": contenido textual, sin descendientes ni atributos. [cadena: Tipo Simple Predefinido xs:string]
  `<xs:element name="distancia" type="xs:string"/>`

- "puntosinteres": elemento con descendientes y sin atributos. [Tipo Complejo, al que denominaremos "tipoPuntosInteres"]
  `<xs:element name="puntosinteres" type="tipoPuntosInteres"/>`

- "punto": elemento con descendientes y con atributos. [Tipo Complejo, al que denominaremos "tipoPunto"]
  `<xs:element name="punto" type="tipoPunto"/>`

- "nombre": contenido textual, sin descendientes ni atributos. [cadena: Tipo Simple Predefinido xs:string]
  `<xs:element name="nombre" type="xs:string"/>`

- "puntokilometrico": contenido textual, sin descendientes ni atributos. [Estimamos que el contenido será numérico, con dos decimales, por lo que al

tipo simple predefinido le vamos a aplicar restricciones: Tipo Simple Construido, al que denominaremos "tipoKm"]

```
<xs:element name="puntokilometrico" type="tipoKm"/>
```

- "coordenadas": contenido textual, sin descendientes ni atributos. [Estimamos que el contenido será una cadena con el formato xxx;xxx, por lo que al tipo simple predefinido le vamos a aplicar restricciones: Tipo Simple Construido, al que denominaremos "tipoCoordenadas"]

```
<xs:element name="coordenadas" type="tipoCoordenadas"/>
```

- "plano": contenido textual, sin descendientes y con atributos. [cadena con atributos: Tipo Complejo, al que denominaremos "tipoPlano"]

```
<xs:element name="plano" type="tipoPlano"/>
```

▫ Esquema XML

```
<xs:element name="ruta" type="tipoERuta"/>

<xs:element name="denominacion" type="tipoCadena"/>
<xs:element name="distancia" type="xs:string"/>
<xs:element name="puntosinteres" type="tipoPuntosInteres"/>
<xs:element name="plano" type="tipoPlano"/>

<xs:element name="punto" type="tipoPunto"/>
<xs:element name="nombre" type="xs:string"/>
<xs:element name="puntokilometrico" type="tipoKm"/>
<xs:element name="coordenadas" type="tipoCoordenadas"/>
```

1.4. Declaración de elementos hijo en el Esquema XML.

A continuación vamos a empezar a desarrollar aquellos tipos de datos para los que un elemento tiene descendientes (<xs:sequence>)
(En el paso anterior hemos detallado los elementos existentes, sin entrar a describir su composición. Para todos los tipos de datos distintos de los simples predefinidos hemos nombrado una denominación que desarrollaremos posteriormente).

Elementos con descendientes: al declarar los elementos descendientes los referenciaremos (ref="..") a los elementos declarados en el paso anterior.

- "ruta": [Descendientes y atributos: Tipo Complejo, "tipoERuta"]

```
<ruta … >
   <denominacion>…</denominacion>
   <distancia>…</distancia>
   <puntosinteres>
        …
   </puntosinteres>
   <plano … >…</plano>
</ruta>
```

Todos los elementos descendientes van a aparecer una única vez.
(Los atributos del elemento "ruta" serán descritos más adelante).

```
<xs:complexType name="tipoERuta">
   <xs:sequence>
      <xs:element ref="denominacion"/>
      <xs:element ref="distancia"/>
```

```
        <xs:element ref="puntosinteres"/>
        <xs:element ref="plano"/>
    </xs:sequence>

</xs:complexType>
```

- "puntosinteres": [Descendientes sin atributos: Tipo Complejo, "tipoPuntosInteres"]

```
  <puntosinteres>
      <punto … >
            …
      </punto>
      …
  </puntosinteres>
```

Solo presenta un descendiente, el elemento "punto", que puede aparecer varias veces. (*se puede detallar el atributo maxOccurs en la etiqueta "sequence" o en la etiqueta del propio elemento. En el caso de que existieran varios elementos hijos, si se especificara en la etiqueta "sequence" implicaría que todo el grupo de descendientes se podría repetir mientras que, si se especificara en un elemento, únicamente sería el que se podría repetir*).

```
 <xs:complexType name="tipoPuntosInteres">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="punto"/>
    </xs:sequence>
 </xs:complexType>
```

- "punto": [Descendientes y atributos: Tipo Complejo, "tipoPunto"]

```
      <punto … >
            <nombre>…</nombre>
            <puntokilometrico>…</puntokilometrico>
      </punto>
      <punto … >
            <nombre>…</nombre>
            <coordenadas>…</coordenadas>
      </punto>
```

El elemento descendiente "nombre" puede aparecer 1 vez. Los elementos "puntokilometrico" y "coordenadas" deben ser excluyentes (uno u el otro), apareciendo 1 vez.

```
 <xs:complexType name="tipoPunto">
    <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:choice>
        <xs:element ref="puntokilometrico"/>
        <xs:element ref="coordenadas"/>
      </xs:choice>
    </xs:sequence>

</xs:complexType>
```

▫ Esquema XML

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

 <!-- Elementos -->
 <xs:element name="ruta" type="tipoERuta"/>

 <xs:element name="denominacion" type="tipoCadena"/>
 <xs:element name="distancia" type="xs:string"/>
 <xs:element name="puntosinteres" type="tipoPuntosInteres"/>
 <xs:element name="plano" type="tipoPlano"/>

 <xs:element name="punto" type="tipoPunto"/>
 <xs:element name="nombre" type="xs:string"/>
 <xs:element name="puntokilometrico" type="tipoKm"/>
 <xs:element name="coordenadas" type="tipoCoordenadas"/>

 <!-- Tipos de Datos (construidos y complejos) -->
 <xs:complexType name="tipoERuta">
    <xs:sequence>
       <xs:element ref="denominacion"/>
       <xs:element ref="distancia"/>
       <xs:element ref="puntosinteres"/>
       <xs:element ref="plano"/>
    </xs:sequence>

 </xs:complexType>

 <xs:complexType name="tipoPuntosInteres">
    <xs:sequence maxOccurs="unbounded">
       <xs:element ref="punto"/>
    </xs:sequence>
 </xs:complexType>

 <xs:complexType name="tipoPunto">
    <xs:sequence>
       <xs:element ref="nombre"/>
       <xs:choice>
          <xs:element ref="puntokilometrico"/>
          <xs:element ref="coordenadas"/>
       </xs:choice>
    </xs:sequence>

 </xs:complexType>

</xs:schema>
```

En este punto hemos declarado todos los elementos y su estructura de árbol.
Mediante este método podríamos reutilizar los elementos descritos.

1.5. Definición de los atributos en el Esquema XML.

Creamos un tipo para la declaración de los atributos (de forma similar a lo desarrollado con los elementos).
Elementos que presentan atributos:

- "ruta": [Descendientes y atributos: Tipo Complejo, "tipoRuta"]

<ruta cod="R01" tiporuta="Montaña" dificultad=”Baja”>

- Atributo “cod”: [tipo ID, cadena con formato letra-dos números: tipo simple construido, denominado “tipoCodRuta”]
  Dentro del tipo “TipoERuta” declaramos que va a tener dicho atributo:
  <xs:attribute ref="cod" use="required"/>
  Posteriormente, de forma independiente, declaramos el atributo:
  <xs:attribute name="cod" type="tipoCodRuta"/>

- Atributo “tiporuta”: [tipo cadena, con lista de valores determinados que puede adoptar: tipo simple construido, denominado “tipoCatRutas”]
  Dentro del tipo “TipoERuta” declaramos que va a tener dicho atributo:
  <xs:attribute ref="tiporuta" use="required"/>
  Posteriormente, de forma independiente, declaramos el atributo:
  <xs:attribute name="tiporuta" type="tipoCatRutas"/>

- Atributo “dificultad”: [tipo cadena, con lista de valores determinados que puede adoptar: tipo simple construido, denominado “tipoDificultad”]
  Dentro del tipo “TipoERuta” declaramos que va a tener dicho atributo:
  <xs:attribute ref="dificultad" use="required"/>
  Posteriormente, de forma independiente, declaramos el atributo:
  <xs:attribute name="dificultad" type="tipoDificultad"/>

  Nota: se podría haber creado un grupo de atributos que englobara a estos tres atributos.

o “punto”: [Descendientes y atributos: Tipo Complejo, “tipoPunto”]
<punto pos="P01">

- Atributo “pos”: [tipo ID, cadena con formato letra P-dos números: tipo simple construido, denominado “tipoPosPunto”]
  Dentro del tipo “TipoPunto” declaramos que va a tener dicho atributo:
  <xs:attribute ref="pos" use="required"/>
  Posteriormente, de forma independiente, declaramos el atributo:
  <xs:attribute name="pos" type="tipoPosPunto"/>

o “plano”: [cadena con atributos: Tipo Complejo, “tipoPlano”]
<plano tipo="JPG">

- Atributo “tipo”: [tipo notación, haciendo referencia a los tipos MIME, denominado “tipoTipoImg”]
  (El atributo “tipo” indica el tipo de archivo MIME de la imagen.)
  Dado que hacemos referencia a códigos MIME habrá que indicar las correspondientes notaciones (aunque el archivo que se indicar en el documento XML es .jpg, añadiremos también la notación al formato .gif).
  <xs:notation name="GIF" system="image/gif"/>
  <xs:notation name="JPG" system="image/jpeg"/>
  Declaramos el tipo “TipoPlano”, que va a tener dicho atributo:
  <xs:complexType name="tipoPlano">

```
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="tipo" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

Posteriormente, de forma independiente, declaramos el atributo:

```
<xs:attribute name="tipo" type="tipoTipoImg"/>
```

▫ Esquema XML

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

  <!-- Notaciones -->
  <xs:notation name="GIF" system="image/gif"/>
  <xs:notation name="JPG" system="image/jpeg"/>

  <!-- Elementos -->
  <xs:element name="ruta" type="tipoERuta"/>

  <xs:element name="denominacion" type="tipoCadena"/>
  <xs:element name="distancia" type="xs:string"/>
  <xs:element name="puntosinteres" type="tipoPuntosInteres"/>
  <xs:element name="plano" type="tipoPlano"/>

  <xs:element name="punto" type="tipoPunto"/>
  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="puntokilometrico" type="tipoKm"/>
  <xs:element name="coordenadas" type="tipoCoordenadas"/>

  <!-- Atributos -->
  <xs:attribute name="cod" type="tipoCodRuta"/>
  <xs:attribute name="tiporuta" type="tipoCatRutas"/>
  <xs:attribute name="dificultad" type="tipoDificultad"/>

  <xs:attribute name="pos" type="tipoPosPunto"/>

  <xs:attribute name="tipo" type="tipoTipoImg"/>

  <!-- Tipos de Datos (construidos y complejos) -->
  <xs:complexType name="tipoERuta">
     <xs:sequence>
        <xs:element ref="denominacion"/>
        <xs:element ref="distancia"/>
        <xs:element ref="puntosinteres"/>
        <xs:element ref="plano"/>
     </xs:sequence>
     <xs:attribute ref="cod" use="required"/>
     <xs:attribute ref="tiporuta" use="required"/>
     <xs:attribute ref="dificultad" use="required"/>
  </xs:complexType>

  <xs:complexType name="tipoPuntosInteres">
     <xs:sequence maxOccurs="unbounded">
        <xs:element ref="punto"/>
     </xs:sequence>
  </xs:complexType>
```

```
<xs:complexType name="tipoPunto">
   <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:choice>
         <xs:element ref="puntokilometrico"/>
         <xs:element ref="coordenadas"/>
      </xs:choice>
   </xs:sequence>
   <xs:attribute ref="pos" use="required"/>
</xs:complexType>

<xs:complexType name="tipoPlano">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="tipo" use="required"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

</xs:schema>
```

En este punto hemos declarado todos los elementos y su estructura de árbol, así como todos los atributos.
Mediante este método podríamos reutilizar los elementos y atributos descritos.

1.6. Definición de los tipos de datos (construidos y complejos) en el Esquema XML.

Finalmente construimos los tipos de datos: tipos de datos simples construidos y tipos de datos complejos.
Repasamos los elementos y atributos declarados e identificamos los tipos de datos a construir:

a) Tipos de datos relacionados con los elementos:

- "tipoERuta": relacionado con el elemento "ruta" (raíz XML).
  Ya construido y finalizado.
- "tipoCadena": relacionado con el elemento "denominacion".
  [Sería del tipo simple predefinido, cadena, pero vamos a indicarle un tipo simple construido para agregarle restricciones (1 a 30 caracteres): Tipo Simple Construido, que denominaremos "tipoCadena"]

```
<xs:simpleType name="tipoCadena">
    <xs:restriction base="xs:string">
       <xs:minLength value="1"/>
       <xs:maxLength value="30"/>
   </xs:restriction>
</xs:simpleType>
```

- "tipoPuntosInteres": relacionado con el elemento "puntosinteres".
  Ya construido y finalizado.
- "tipoPlano": relacionado con el elemento "plano".
  Debido a que contiene contenido textual y atributos (o elementos) indicamos el atributo "mixed="true"" en la descripción del tipo.

```
<xs:complexType name="tipoPlano" mixed="true">
```

- "tipoPunto": relacionado con el elemento "punto".
  Ya construido y finalizado.

- "tipoKm": relacionado con el elemento "puntokilometrico".
[Estimamos que el contenido será numérico, con dos decimales, por lo que al tipo simple predefinido le vamos a aplicar restricciones: Tipo Simple Construido, al que denominaremos "tipoKm"]
(Importante: al definir el tipo de dato numérico con dos decimales, la información debe venir con el punto "." y no la coma "," como delimitador de los decimales en el documento XML)

```
<xs:simpleType name="tipoKm">
   <xs:restriction base="xs:decimal">
      <xs:minInclusive value="0"/>
      <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>
```

- "tipoCoordenadas": relacionado con el elemento "coordenadas".
[Estimamos que el contenido será una cadena con el formato xxx;xxx, por lo que al tipo simple predefinido le vamos a aplicar restricciones: Tipo Simple Construido, al que denominaremos "tipoCoordenadas"]

```
<xs:simpleType name="tipoCoordenadas">
   <xs:restriction base="xs:string">
      <xs:maxLength value="7"/>
      <xs:pattern value="[0-9]{0,3}[;][0-9]{0,3}"/>
   </xs:restriction>
</xs:simpleType>
```

b) Tipos de datos relacionados con los atributos:

- "tipoCodRuta": relacionado con el atributo "cod" ("ruta").
[tipo ID, cadena con formato letra-dos números: tipo simple construido, denominado "tipoCodRuta"]

```
<xs:simpleType name="tipoCodRuta">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[A-Z][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>
```

- tipoCatRutas: relacionado con el atributo "tiporuta" ("ruta").
[tipo cadena, con lista de valores determinados que puede adoptar: tipo simple construido, denominado "tipoCatRutas"]

```
<xs:simpleType name="tipoCatRutas">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Montaña"/>
      <xs:enumeration value="Ciclista"/>
      <xs:enumeration value="Monumental"/>
   </xs:restriction>
</xs:simpleType>
```

- tipoDificultad: relacionado con el atributo "dificultad" ("ruta").
[tipo cadena, con lista de valores determinados que puede adoptar: tipo simple construido, denominado "tipoDificultad"]

```
<xs:simpleType name="tipoDificultad">
   <xs:restriction base="xs:string">
      <xs:enumeration value="Alta"/>
      <xs:enumeration value="Media"/>
```

```
        <xs:enumeration value="Baja"/>
    </xs:restriction>
</xs:simpleType>
```

- tipoPosPunto: relacionado con el atributo "pos" ("punto"). [tipo ID, cadena con formato letra P-dos números: tipo simple construido, denominado "tipoPosPunto"]

```
<xs:simpleType name="tipoPosPunto">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[P][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>
```

- tipoTipoImg: relacionado con el atributo "tipo" ("plano"). [tipo notación, haciendo referencia a los tipos MIME, denominado "tipoTipoImg"]

```
<xs:simpleType name="tipoTipoImg">
    <xs:restriction base="xs:string">
        <xs:enumeration value="GIF"/>
        <xs:enumeration value="JPG"/>
    </xs:restriction>
</xs:simpleType>
```

▫ Documento XML: "ruta01.xml"

```
<?xml version="1.0" encoding="UTF-8"?>
<ruta xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="rutas.xsd"
    cod="R01" tiporuta="Montaña" dificultad="Baja">
  <denominacion>Ruta de las Cascadas</denominacion>
  <distancia>8 Km</distancia>
  <puntosinteres>
      <punto pos="P01">
        <nombre>Cascada 1</nombre>
        <puntokilometrico>1.5</puntokilometrico>
      </punto>
      <punto pos="P02">
        <nombre>Cascada 2</nombre>
        <puntokilometrico>2.3</puntokilometrico>
      </punto>
      <punto pos="P03">
        <nombre>Cascada 3</nombre>
        <coordenadas>259;367</coordenadas>
      </punto>
  </puntosinteres>
  <plano tipo="JPG">MapaR01.jpg</plano>
</ruta>
```

▫ Esquema XML: "rutas.xsd"

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

```
<!-- Notaciones -->
<xs:notation name="GIF" system="image/gif"/>
<xs:notation name="JPG" system="image/jpeg"/>

<!-- Elementos -->
<xs:element name="ruta" type="tipoERuta"/>

<xs:element name="denominacion" type="tipoCadena"/>
<xs:element name="distancia" type="xs:string"/>
<xs:element name="puntosinteres" type="tipoPuntosInteres"/>
<xs:element name="plano" type="tipoPlano"/>

<xs:element name="punto" type="tipoPunto"/>
<xs:element name="nombre" type="xs:string"/>
<xs:element name="puntokilometrico" type="tipoKm"/>
<xs:element name="coordenadas" type="tipoCoordenadas"/>

<!-- Atributos -->
<xs:attribute name="cod" type="tipoCodRuta"/>
<xs:attribute name="tiporuta" type="tipoCatRutas"/>
<xs:attribute name="dificultad" type="tipoDificultad"/>

<xs:attribute name="pos" type="tipoPosPunto"/>

<xs:attribute name="tipo" type="tipoTipoImg"/>

<!-- Tipos de Datos (construidos y complejos) -->
<xs:complexType name="tipoERuta">
   <xs:sequence>
      <xs:element ref="denominacion"/>
      <xs:element ref="distancia"/>
      <xs:element ref="puntosinteres"/>
      <xs:element ref="plano"/>
   </xs:sequence>
   <xs:attribute ref="cod" use="required"/>
   <xs:attribute ref="tiporuta" use="required"/>
   <xs:attribute ref="dificultad" use="required"/>
</xs:complexType>

<xs:complexType name="tipoPuntosInteres">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="punto"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoPunto">
   <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:choice>
         <xs:element ref="puntokilometrico"/>
         <xs:element ref="coordenadas"/>
      </xs:choice>
   </xs:sequence>
   <xs:attribute ref="pos" use="required"/>
</xs:complexType>

<xs:complexType name="tipoPlano" mixed=”true”>
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="tipo" use="required"/>
```

```
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="tipoKm">
    <xs:restriction base="xs:decimal">
        <xs:minInclusive value="0"/>
        <xs:fractionDigits value="2"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoCoordenadas">
    <xs:restriction base="xs:string">
        <xs:maxLength value="7"/>
        <xs:pattern value="[0-9]{0,3}[;][0-9]{0,3}"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoCadena">
    <xs:restriction base="xs:string">
        <xs:minLength value="1"/>
        <xs:maxLength value="30"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoCodRuta">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[A-Z][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoCatRutas">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Montaña"/>
        <xs:enumeration value="Ciclista"/>
        <xs:enumeration value="Monumental"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoDificultad">
    <xs:restriction base="xs:string">
        <xs:enumeration value="Alta"/>
        <xs:enumeration value="Media"/>
        <xs:enumeration value="Baja"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosPunto">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[P][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoTipoImg">
    <xs:restriction base="xs:string">
        <xs:enumeration value="GIF"/>
        <xs:enumeration value="JPG"/>
    </xs:restriction>
```

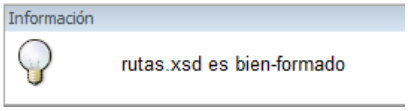
```
 </xs:simpleType>

</xs:schema>
```

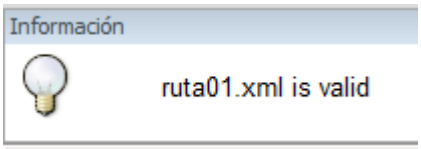
(Nota: se ha desarrollado la especificación de tipos de forma exhaustiva. En un esquema convencional no es necesario, aunque si recomendable, efectuarlo con tanta profundidad. Este desarrollo permite la reutilización de todos los tipos, tanto en este esquema, como con cualquier otro que se desee vincular).

(Se ha incorporado al ejercicio numerosas restricciones sobre la información contenida en la instancia (documento XML), a diferencia de las posibilidades que ofrece la validación a través de DTD. Por ello el esquema se hace más extenso).

Finalmente se verificará que no existen errores en el Esquema XML (cotejar con documento XML) y se podría comprobar que está bien formado (comprobación de sintaxis).



Posteriormente se puede validar el documento XML con el Esquema XML asociado, siendo totalmente válido.



El documento XML “ruta01.xml” está bien formado y es válido, conforme a las especificaciones contenidas en el Esquema XML “rutas.xsd”.

▫ Otra posible resolución siguiendo el modelo de diseño Plano (válida también, pero con menos exhaustividad en la declaración de tipos):

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

 <xs:notation name="GIF" system="image/gif"/>
 <xs:notation name="JPG" system="image/jpeg"/>

 <xs:element name="ruta">
  <xs:complexType>
   <xs:sequence>
    <xs:element ref="denominacion"/>
    <xs:element ref="distancia"/>
    <xs:element ref="puntosinteres"/>
    <xs:element ref="plano"/>
   </xs:sequence>
   <xs:attribute name="cod" type="xs:ID" use="required"/>
   <xs:attribute name="tiporuta" use="required">
    <xs:simpleType>
     <xs:restriction base="xs:string">
      <xs:enumeration value="Montaña"/>
      <xs:enumeration value="Ciclista"/>
      <xs:enumeration value="Monumental"/>
     </xs:restriction>
    </xs:simpleType>
```

```
    </xs:attribute>
    <xs:attribute name="dificultad" use="required">
     <xs:simpleType>
      <xs:restriction base="xs:string">
       <xs:enumeration value="Alta"/>
       <xs:enumeration value="Media"/>
       <xs:enumeration value="Baja"/>
      </xs:restriction>
     </xs:simpleType>
    </xs:attribute>
   </xs:complexType>
  </xs:element>

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

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

  <xs:element name="puntosinteres">
   <xs:complexType>
    <xs:sequence maxOccurs="unbounded">
     <xs:element ref="punto"/>
    </xs:sequence>
   </xs:complexType>
  </xs:element>

  <xs:element name="plano">
   <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute name="tipo" use="required">
          <xs:simpleType>
            <xs:restriction base="xs:string">
              <xs:enumeration value="GIF"/>
              <xs:enumeration value="JPG"/>
            </xs:restriction>
          </xs:simpleType>
        </xs:attribute>
      </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
  </xs:element>

  <xs:element name="punto">
   <xs:complexType>
    <xs:sequence>
     <xs:element ref="nombre"/>
     <xs:choice>
      <xs:element ref="puntokilometrico"/>
      <xs:element ref="coordenadas"/>
     </xs:choice>
    </xs:sequence>
    <xs:attribute name="pos" type="xs:ID" use="required"/>
   </xs:complexType>
  </xs:element>

  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="puntokilometrico" type="xs:string"/>
  <xs:element name="coordenadas" type="xs:string"/>

</xs:schema>
```

2. Ejercicio "EQUIPO"

Se desea generar una estructura de archivo, en formato XML, para que cada equipo que participa en alguna liga comunique a la Federación sus integrantes.
(El presente ejercicio comienza desde una estructura simple y, en cada apartado, se van añadiendo componentes con objeto de facilitar su comprensión y elaboración).

► Se pide: generar el documento XML "equipo_partido.xml" y el esquema XML "equipo.xsd", siguiendo las instrucciones que se detallan a continuación. Comprobar que está bien formado y validar.

2.1. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo>
  <jugadores>Yo</jugadores>
</equipo>
```

Construir su correspondiente Esquema XML, en documento externo privado, asociar al documento XML y validar.
El elemento "jugadores" tiene que aparecer 1 vez.

▫ Asociación documento XML con Esquema XML:
En el propio documento XML insertamos la línea que lo asocia al esquema XML que se utilizará para su valiación.

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

▫ Esquema XML:
El documento XML solo consta de dos elementos:

- *"equipo": nodo raíz. Elemento sin contenido textual ni atributos. Solo consta de descendientes [Tipo Complejo]*
- *"jugadores": (en este caso una única ocurrencia, y contiene texto [Tipo simple]).*

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

 <xs:element name="equipo">
   <xs:complexType>
     <xs:sequence>
       <xs:element name="jugadores" type="xs:string"/>
     </xs:sequence>
   </xs:complexType>
 </xs:element>

</xs:schema>
```

2.2. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
   denominacion="MiEquipo">
  <jugadores>Yo</jugadores>
</equipo>
```

Construir su correspondiente Esquema XML, en documento externo privado, y validar.
Se ha añadido al elemento "equipo" (nodo raíz) el atributo "denominación", que es obligatorio.

- Atributo "denominación": tipo cadena [Tipo Simple] y requerido. Se agrega dentro del tipo de su elemento "equipo" [Tipo Complejo]

▫ Esquema XML:

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

  <xs:element name="equipo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="jugadores" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="denominacion" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

2.3. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1">
  <jugadores>Yo</jugadores>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
También se ha añadido al elemento "equipo" (nodo raíz) el atributo "categoría", que es obligatorio.

- Atributo "categoria": tipo cadena [Tipo Simple] y requerido. Se agrega dentro del tipo de su elemento "equipo" [Tipo Complejo]

▫ Esquema XML:

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

  <xs:element name="equipo">
```

```
    <xs:complexType>
      <xs:sequence>
        <xs:element name="jugadores" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="denominacion" type="xs:string" use="required"/>
      <xs:attribute name="categoria" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

2.4. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1">
  <jugadores>Yo</jugadores>
  <tecnicos>Tú</tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Se añade el elemento "técnicos" al elemento "equipo". Ahora el elemento "equipo" está compuesto de los elementos "jugadores" y "técnicos". El elemento "tecnicos" tiene que aparecer 1 vez, a continuación del elemento "jugadores".

▫ Esquema XML:

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

  <xs:element name="equipo">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="jugadores" type="xs:string"/>
        <xs:element name="tecnicos" type="xs:string"/>
      </xs:sequence>
      <xs:attribute name="denominacion" type="xs:string" use="required"/>
      <xs:attribute name="categoria" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>

</xs:schema>
```

Con el propósito de mejorar la comprensión y desarrollo del Esquema XML vamos a crear el "tipoEquipo" de forma autónoma (esto también favorecerá el desarrollo posterior del esquema, así como la reutilización).
(El código sombreado en azul más oscuro es el tipo de datos que vamos a extraer para su declaración autónoma).

▫ Esquema XML:

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

 <!-- Elementos -->
 <xs:element name="equipo" type="tipoEquipo"/>

 <!-- Tipos de Datos (construidos y complejos) -->
 <xs:complexType name="tipoEquipo">
    <xs:sequence>
       <xs:element name="jugadores" type="xs:string"/>
       <xs:element name="tecnicos" type="xs:string"/>
    </xs:sequence>
    <xs:attribute name="denominacion" type="xs:string" use="required"/>
    <xs:attribute name="categoria" type="xs:string" use="required"/>
 </xs:complexType>

</xs:schema>
```

2.5. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1">
  <jugadores>
       <jugador>Él</jugador>
       <jugador>Ella</jugador>
  </jugadores>
  <tecnicos>Tú</tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
El elemento “jugadores” está compuesto del elemento “jugador”, que puede aparecer 1 o varias veces.
(A partir de este momento vamos a declarar los elementos, atributos y tipos de forma independiente, no de forma anidada).

- “jugadores”: elemento con descendientes [Tipo Complejo]. Por dicho motivo sustituimos el tipo cadena declarado hasta ahora por el tipo complejo.
  Declaramos el elemento “jugadores”:

```
<xs:element name="jugadores" type="tipoJugadores"/>
```

  Modificamos el “tipoEquipo”:

```
<xs:element name="jugadores" type="xs:string"/>
```

```
<xs:element ref="jugadores"/>
```

  Declaramos el tipo de datos “tipoJugadores”: contiene el elemento “jugador” con múltiples ocurrencias

```
 <xs:complexType name="tipoJugadores">
     <xs:sequence maxOccurs="unbounded">
        <xs:element ref="jugador"/>
     </xs:sequence>
```

```
  </xs:complexType>
```

- "jugador": tipo cadena [Tipo simple predefinido]

```
  <xs:element name="jugador" type="xs:string"/>
```

▫ Esquema XML:

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

  <!-- Elementos -->
  <xs:element name="equipo" type="tipoEquipo"/>

  <xs:element name="jugadores" type="tipoJugadores"/>
  <xs:element name="jugador" type="xs:string"/>

  <!-- Tipos de Datos (construidos y complejos) -->
  <xs:complexType name="tipoEquipo">
     <xs:sequence>
        <xs:element ref="jugadores"/>
        <xs:element name="tecnicos" type="xs:string"/>
     </xs:sequence>
     <xs:attribute name="denominacion" type="xs:string" use="required"/>
     <xs:attribute name="categoria" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="tipoJugadores">
     <xs:sequence maxOccurs="unbounded">
        <xs:element ref="jugador"/>
     </xs:sequence>
  </xs:complexType>

</xs:schema>
```

Vemos que mediante esta metodología vamos declarando los elementos, los atributos y los tipos de datos (excepto los simples predefinidos) de forma independiente y referenciándolos según su uso.
Esto facilita el desarrollo, el seguimiento, el reaprovechamiento, las modificaciones, etc.

2.6. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1">
  <jugadores>
       <jugador dorsal="D01" posicion="base">Él</jugador>
       <jugador dorsal="D02">Ella</jugador>
  </jugadores>
  <tecnicos>Tú</tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.

El elemento “jugador” tiene dos atributos: (Convertimos el tipo de datos del elemento “jugador” de tipo cadena a tipo complejo)

- El atributo “dorsal” será obligatorio, correspondiendo con el identificador único del jugador en el equipo.

```
<xs:attribute name="dorsal" type="tipoDorsal"/>
```

Declaramos su tipo (añadimos restricciones a tipo cadena [Tipo Construido])

```
<xs:simpleType name="tipoDorsal">
  <xs:restriction base="xs:ID">
    <xs:length value="3"/>
    <xs:pattern value="[D][0-9][0-9]"/>
  </xs:restriction>
</xs:simpleType>
```

Creamos el “tipoJugador” (anteriormente era simplemente tipo cadena y ahora es un tipo Complejo compuesto de cadena y atributos):

```
<xs:complexType name="tipoJugador" mixed="true">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="dorsal" use="required"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Modificamos el elemento “jugador” asociándolo el nuevo tipo:

```
<xs:element name="jugador" type="tipoJugador"/>
```

- El atributo “posicion” será opcional y, en caso de ser indicado, adoptará un solo valor de la siguiente relación: base, escolta, alero, alapivot, pivot. (si no se especifica este atributo quedará vacío).

Creamos el atributo “posición”

```
<xs:attribute name="posicion" type="tipoPosicion"/>
```

Declaramos su tipo (contiene enumeración de valores):

```
<xs:simpleType name="tipoPosicion">
  <xs:restriction base="xs:string">
    <xs:enumeration value="base"/>
    <xs:enumeration value="escolta"/>
    <xs:enumeration value="alero"/>
    <xs:enumeration value="alapivot"/>
    <xs:enumeration value="pivot"/>
  </xs:restriction>
</xs:simpleType>
```

Asociamos el atributo al “tipoJugador”:

```
<xs:attribute ref="posicion" use="optional"/>
```

(No sería necesario añadir el indicativo “optional” dado que es el valor por defecto)

▫ Esquema XML:

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

 <!-- Elementos -->
 <xs:element name="equipo" type="tipoEquipo"/>
```

```
<xs:element name="jugadores" type="tipoJugadores"/>
<xs:element name="jugador" type="tipoJugador"/>

<!-- Atributos -->
<xs:attribute name="dorsal" type="tipoDorsal"/>
<xs:attribute name="posicion" type="tipoPosicion"/>

<!-- Tipos de Datos (construidos y complejos) -->
<xs:complexType name="tipoEquipo">
   <xs:sequence>
      <xs:element ref="jugadores"/>
      <xs:element name="tecnicos" type="xs:string"/>
   </xs:sequence>
   <xs:attribute name="denominacion" type="xs:string" use="required"/>
   <xs:attribute name="categoria" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="tipoJugadores">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="jugador"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoJugador" mixed="true">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="dorsal" use="required"/>
         <xs:attribute ref="posicion" use="optional"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="tipoDorsal">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[D][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
   <xs:restriction base="xs:string">
      <xs:enumeration value="base"/>
      <xs:enumeration value="escolta"/>
      <xs:enumeration value="alero"/>
      <xs:enumeration value="alapivot"/>
      <xs:enumeration value="pivot"/>
   </xs:restriction>
</xs:simpleType>

</xs:schema>
```

2.7. Dado el siguiente documento XML, bien formado

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

```
    xs:noNamespaceSchemaLocation="equipo.xsd"
    denominacion="MiEquipo" categoria="Liga1">
 <jugadores>
     <jugador dorsal="D01" posicion="base">Él</jugador>
     <jugador dorsal="D02">Ella</jugador>
 </jugadores>
 <tecnicos>
     <tecnico codigo="T01" cargo="entrenador">Tú</tecnico>
     <tecnico codigo="T02" cargo="fisio">Tuya</tecnico>
 </tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
(Las modificaciones y añadidos a efectuar son similares a lo realizado en los dos apartados anteriores con los elementos "jugadores" y "jugador").
El elemento "tecnicos" está compuesto del elemento "tecnico", que puede aparecer 1 o varias veces.
Declaramos el elemento "tecnicos":

```
<xs:element name="tecnicos" type="tipoTecnicos"/>
```

Declaramos el "tipoTecnicos":

```
<xs:complexType name="tipoTecnicos">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tecnico"/>
   </xs:sequence>
</xs:complexType>
```

Modificamos el "tipoEquipo":
~~<xs:element name="tecnicos" type="xs:string"/>~~

```
<xs:element ref="tecnicos"/>
```

El elemento "tecnico" tiene dos atributos:
- El atributo "codigo" será obligatorio, correspondiendo con el identificador único del técnico en el equipo.
- El atributo "cargo" será obligatorio y adoptará un solo valor sin espacios ni símbolos especiales.

Declaramos el elemento "tecnico":

```
<xs:element name="tecnico" type="tipoTecnico"/>
```

Declaramos el "tipoTecnico", con contenido textual y dos atributos:

```
<xs:complexType name="tipoTecnico" mixed="true">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="codigo" use="required"/>
         <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>
```

Declaramos el atributo "codigo":

```
<xs:attribute name="codigo" type="tipoCodigoTecnico"/>
```

Declaramos el “tipoCodigoTecnico”:

```
<xs:simpleType name="tipoCodigoTecnico">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[T][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>
```

(podríamos haber aprovechado el mismo tipo que el dorsal del jugador, pero existe diferencia en cuanto a la letra del código, por lo que hacemos una declaración específica para cada uno de ellos).

Declaramos el atributo “cargo”: *(el tipo de dato es simple predefinido, por lo que no hay que declarar ningún tipo de dato construido o complejo)*

```
<xs:attribute name="cargo" type="xs:NMTOKEN"/>
```

- Esquema XML:

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

  <!-- Elementos -->
  <xs:element name="equipo" type="tipoEquipo"/>

  <xs:element name="jugadores" type="tipoJugadores"/>
  <xs:element name="jugador" type="tipoJugador"/>

  <xs:element name="tecnicos" type="tipoTecnicos"/>
  <xs:element name="tecnico" type="tipoTecnico"/>

  <!-- Atributos -->
  <xs:attribute name="dorsal" type="tipoDorsal"/>
  <xs:attribute name="posicion" type="tipoPosicion"/>

  <xs:attribute name="codigo" type="tipoCodigoTecnico"/>
  <xs:attribute name="cargo" type="xs:NMTOKEN"/>

  <!-- Tipos de Datos (construidos y complejos) -->
  <xs:complexType name="tipoEquipo">
     <xs:sequence>
        <xs:element ref="jugadores"/>
        <xs:element ref="tecnicos"/>
     </xs:sequence>
     <xs:attribute name="denominacion" type="xs:string" use="required"/>
     <xs:attribute name="categoria" type="xs:string" use="required"/>
  </xs:complexType>

  <xs:complexType name="tipoJugadores">
     <xs:sequence maxOccurs="unbounded">
        <xs:element ref="jugador"/>
     </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tipoJugador" mixed="true">
     <xs:simpleContent>
        <xs:extension base="xs:string">
           <xs:attribute ref="dorsal" use="required"/>
           <xs:attribute ref="posicion" use="optional"/>
        </xs:extension>
```

```
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="tipoDorsal">
    <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[D][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="tipoPosicion">
    <xs:restriction base="xs:string">
      <xs:enumeration value="base"/>
      <xs:enumeration value="escolta"/>
      <xs:enumeration value="alero"/>
      <xs:enumeration value="alapivot"/>
      <xs:enumeration value="pivot"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tecnico"/>
    </xs:sequence>
  </xs:complexType>

  <xs:complexType name="tipoTecnico" mixed="true">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="codigo" use="required"/>
        <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="tipoCodigoTecnico">
    <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[T][0-9][0-9]"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

2.8. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="equipo.xsd"
    denominacion="MiEquipo" categoria="Liga1">
 <jugadores>
      <jugador dorsal="D01" posicion="base">
        <nombre>Jugador1</nombre>
        <apellidos>Primero</apellidos>
        <nacionalidad>Estadounidense</nacionalidad>
      </jugador>
      <jugador dorsal="D02">
```

```
        <nombre>Jugador2</nombre>
        <apellidos>Segundo</apellidos>
    </jugador>
 </jugadores>
 <tecnicos>
    <tecnico codigo="T01" cargo="entrenador">
        <nombre>Técnico1</nombre>
        <apellidos>Ape1</apellidos>
        <nacionalidad>Española</nacionalidad>
    </tecnico>
    <tecnico codigo="T02" cargo="fisio">
        <nombre>Técnico2</nombre>
        <apellidos>Ape2</apellidos>
    </tecnico>
 </tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Los elementos "jugador" y "tecnico" están compuestos de los elementos:

- "nombre": tiene que aparecer 1 vez, contenido texto.
- "apellidos": tiene que aparecer 1 vez, contenido texto.
- "nacionalidad": puede aparecer 1 o 0 veces, contenido texto.

(Nota: debido a que la estructura para los dos elementos "jugador" y "técnico" es la misma, podemos aprovechar a crear un tipo de datos complejo denominado "datosIntegrantes" que contenga la estructura y hacer referencia a ella evitando la redundancia).

Realizamos la declaración del tipo "datosIntegrantes":

```
<xs:complexType name="datosIntegrantes">
  <xs:sequence>
    <xs:element ref="nombre"/>
    <xs:element ref="apellidos"/>
    <xs:element ref="nacionalidad" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Realizamos las declaraciones de los elementos:

```
<xs:element name="nombre" type="xs:string"/>
<xs:element name="apellidos" type="xs:string"/>
<xs:element name="nacionalidad" type="xs:string"/>
```

Modificamos las declaraciones de los tipos "tipoJugador" y "tipoTecnico": extendemos el tipo "datosIntegrantes" (en lugar de cadena) añadiendo los atributos propios de cada uno de ellos. (su contenido pasa de simple a complejo)
(eliminamos la referencia a tipo de contenido mixto, dado que ya no contiene contenido textual)

```
<xs:complexType name="tipoJugador">
  <xs:complexContent>
    <xs:extension base="datosIntegrantes">
      <xs:attribute ref="dorsal" use="required"/>
      <xs:attribute ref="posicion" use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```
<xs:complexType name="tipoTecnico">
   <xs:complexContent>
      <xs:extension base="datosIntegrantes">
         <xs:attribute ref="codigo" use="required"/>
         <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>
```

- Esquema XML:

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

 <!-- Elementos -->
 <xs:element name="equipo" type="tipoEquipo"/>

 <xs:element name="jugadores" type="tipoJugadores"/>
 <xs:element name="jugador" type="tipoJugador"/>

 <xs:element name="tecnicos" type="tipoTecnicos"/>
 <xs:element name="tecnico" type="tipoTecnico"/>

 <xs:element name="nombre" type="xs:string"/>
 <xs:element name="apellidos" type="xs:string"/>
 <xs:element name="nacionalidad" type="xs:string"/>

 <!-- Atributos -->
 <xs:attribute name="dorsal" type="tipoDorsal"/>
 <xs:attribute name="posicion" type="tipoPosicion"/>

 <xs:attribute name="codigo" type="tipoCodigoTecnico"/>
 <xs:attribute name="cargo" type="xs:NMTOKEN"/>

 <!-- Tipos de Datos (construidos y complejos) -->
 <xs:complexType name="tipoEquipo">
    <xs:sequence>
       <xs:element ref="jugadores"/>
       <xs:element ref="tecnicos"/>
    </xs:sequence>
    <xs:attribute name="denominacion" type="xs:string" use="required"/>
    <xs:attribute name="categoria" type="xs:string" use="required"/>
 </xs:complexType>

 <xs:complexType name="tipoJugadores">
    <xs:sequence maxOccurs="unbounded">
       <xs:element ref="jugador"/>
    </xs:sequence>
 </xs:complexType>

 <xs:complexType name="tipoJugador">
    <xs:complexContent>
       <xs:extension base="datosIntegrantes">
          <xs:attribute ref="dorsal" use="required"/>
          <xs:attribute ref="posicion" use="optional"/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>
```

```
<xs:simpleType name="tipoDorsal">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[D][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
   <xs:restriction base="xs:string">
      <xs:enumeration value="base"/>
      <xs:enumeration value="escolta"/>
      <xs:enumeration value="alero"/>
      <xs:enumeration value="alapivot"/>
      <xs:enumeration value="pivot"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoTecnicos">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tecnico"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoTecnico">
   <xs:complexContent>
      <xs:extension base="datosIntegrantes">
         <xs:attribute ref="codigo" use="required"/>
         <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[T][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>

  <xs:complexType name="datosIntegrantes">
   <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:element ref="apellidos"/>
      <xs:element ref="nacionalidad" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

</xs:schema>
```

2.9. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="equipo.xsd"
    denominacion="MiEquipo" categoria="Liga1">
 <jugadores>
      <jugador dorsal="D01" posicion="base">
```

```
        <nombre>Jugador1</nombre>
        <apellidos>Primero</apellidos>
        <nacionalidad>Estadounidense</nacionalidad>
    </jugador>
    <jugador dorsal="D02">
        <nombre>Jugador2</nombre>
        <apellidos>Segundo</apellidos>
        <edad>21</edad>
    </jugador>
 </jugadores>
 <tecnicos>
    <tecnico codigo="T01" cargo="entrenador">
        <nombre>Técnico1</nombre>
        <apellidos>Ape1</apellidos>
        <nacionalidad>Española</nacionalidad>
    </tecnico>
    <tecnico codigo="T02" cargo="fisio">
        <nombre>Técnico2</nombre>
        <apellidos>Ape2</apellidos>
    </tecnico>
 </tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Se ha añadido al elemento "jugador", dentro de su composición, el elemento "edad", que puede aparecer 1 o 0 veces, contenido texto.

Declaramos el elemento:

```
<xs:element name="edad" type="tipoEdad"/>
```

(Creamos un tipo edad para añadirle restricciones, aunque asignando un tipo simple predefinido positiveInteger ya estaríamos estableciendo que fuese entero >0).

```
 <xs:simpleType name="tipoEdad">
    <xs:restriction base="xs:positiveInteger">
       <xs:maxInclusive value="100"/>
    </xs:restriction>
 </xs:simpleType>
```

Dado que el elemento edad solo afecta al elemento "jugador" no lo añadimos en el tipo "datosIntegrantes" (entonces también se le podría añadir a "tecnico") sino que se lo añadimos al "tipoJugador"

```
        <xs:extension base="datosIntegrantes">
           <xs:sequence>
              <xs:element ref="edad" minOccurs="0"/>
           </xs:sequence>
           …
        </xs:extension>
```

- Esquema XML:

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

```
<!-- Elementos -->
<xs:element name="equipo" type="tipoEquipo"/>

<xs:element name="jugadores" type="tipoJugadores"/>
<xs:element name="jugador" type="tipoJugador"/>

<xs:element name="tecnicos" type="tipoTecnicos"/>
<xs:element name="tecnico" type="tipoTecnico"/>

<xs:element name="nombre" type="xs:string"/>
<xs:element name="apellidos" type="xs:string"/>
<xs:element name="nacionalidad" type="xs:string"/>
<xs:element name="edad" type="tipoEdad"/>

<!-- Atributos -->
<xs:attribute name="dorsal" type="tipoDorsal"/>
<xs:attribute name="posicion" type="tipoPosicion"/>

<xs:attribute name="codigo" type="tipoCodigoTecnico"/>
<xs:attribute name="cargo" type="xs:NMTOKEN"/>

<!-- Tipos de Datos (construidos y complejos) -->
<xs:complexType name="tipoEquipo">
   <xs:sequence>
      <xs:element ref="jugadores"/>
      <xs:element ref="tecnicos"/>
   </xs:sequence>
   <xs:attribute name="denominacion" type="xs:string" use="required"/>
   <xs:attribute name="categoria" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="tipoJugadores">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="jugador"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoJugador">
   <xs:complexContent>
      <xs:extension base="datosIntegrantes">
         <xs:sequence>
            <xs:element ref="edad" minOccurs="0"/>
         </xs:sequence>
         <xs:attribute ref="dorsal" use="required"/>
         <xs:attribute ref="posicion" use="optional"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoDorsal">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[D][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
   <xs:restriction base="xs:string">
      <xs:enumeration value="base"/>
```

```
        <xs:enumeration value="escolta"/>
        <xs:enumeration value="alero"/>
        <xs:enumeration value="alapivot"/>
        <xs:enumeration value="pivot"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="tecnico"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoTecnico">
    <xs:complexContent>
        <xs:extension base="datosIntegrantes">
            <xs:attribute ref="codigo" use="required"/>
            <xs:attribute ref="cargo" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[T][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

  <xs:complexType name="datosIntegrantes">
    <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="apellidos"/>
        <xs:element ref="nacionalidad" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="tipoEdad">
    <xs:restriction base="xs:positiveInteger">
        <xs:maxInclusive value="100"/>
    </xs:restriction>
</xs:simpleType>

</xs:schema>
```

2.10. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1">
 <jugadores>
       <jugador dorsal="D01" posicion="base">
          <nombre>Jugador1</nombre>
          <apellidos>Primero</apellidos>
          <nacionalidad>Estadounidense</nacionalidad>
          <foto formato="JPG">FotoD01.jpg</foto>
```

```
    </jugador>
    <jugador dorsal="D02">
      <nombre>Jugador2</nombre>
      <apellidos>Segundo</apellidos>
      <edad>21</edad>
      <foto formato="GIF">FotoD02.jpg</foto>
    </jugador>
  </jugadores>
  <tecnicos>
    <tecnico codigo="T01" cargo="entrenador">
      <nombre>Técnico1</nombre>
      <apellidos>Ape1</apellidos>
      <nacionalidad>Española</nacionalidad>
    </tecnico>
    <tecnico codigo="T02" cargo="fisio">
      <nombre>Técnico2</nombre>
      <apellidos>Ape2</apellidos>
    </tecnico>
  </tecnicos>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Se le ha añadido al elemento "jugador" el elemento foto, que aparecerá una vez por cada jugador.
El elemento foto contendrá la ruta de la imagen del jugador y el atributo "formato", que indicará el formato de la imagen, pudiendo ser "gif" o "jpg". El atributo será obligatorio.

Declaración del elemento "foto":

```
<xs:element name="foto" type="tipoFoto"/>
```

Declaración del "tipoFoto": (contenido textual y atributo)

```
  <xs:complexType name="tipoFoto" mixed="true">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="formato" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
```

Declaración del atributo "formato":

```
<xs:attribute name="formato" type="tipoFormatoImagen"/>
```

Declaración del "tipoFormatoImagen":

```
<xs:simpleType name="tipoFormatoImagen">
    <xs:restriction base="xs:string">
      <xs:enumeration value="GIF"/>
      <xs:enumeration value="JPG"/>
    </xs:restriction>
  </xs:simpleType>
```

(Nota: para la referenciación al formato de imagen se hace uso de notaciones, para relacionar el código MIME).
Declaraciones de las notaciones códigos MIME:

```
<xs:notation name="GIF" system="image/gif"/>
<xs:notation name="JPG" system="image/jpeg"/>
```

Añadir el elemento “foto” al “tipoJugador”:

```
<xs:sequence>
   <xs:element ref="edad" minOccurs="0"/>
   <xs:element ref="foto"/>
</xs:sequence>
```

▫ Esquema XML:

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

 <!-- Notaciones -->
 <xs:notation name="GIF" system="image/gif"/>
 <xs:notation name="JPG" system="image/jpeg"/>

 <!-- Elementos -->
 <xs:element name="equipo" type="tipoEquipo"/>

 <xs:element name="jugadores" type="tipoJugadores"/>
 <xs:element name="jugador" type="tipoJugador"/>

 <xs:element name="tecnicos" type="tipoTecnicos"/>
 <xs:element name="tecnico" type="tipoTecnico"/>

 <xs:element name="nombre" type="xs:string"/>
 <xs:element name="apellidos" type="xs:string"/>
 <xs:element name="nacionalidad" type="xs:string"/>
 <xs:element name="edad" type="tipoEdad"/>
 <xs:element name="foto" type="tipoFoto"/>

 <!-- Atributos -->
 <xs:attribute name="dorsal" type="tipoDorsal"/>
 <xs:attribute name="posicion" type="tipoPosicion"/>

 <xs:attribute name="codigo" type="tipoCodigoTecnico"/>
 <xs:attribute name="cargo" type="xs:NMTOKEN"/>

 <xs:attribute name="formato" type="tipoFormatoImagen"/>

 <!-- Tipos de Datos (construidos y complejos) -->
 <xs:complexType name="tipoEquipo">
    <xs:sequence>
       <xs:element ref="jugadores"/>
       <xs:element ref="tecnicos"/>
    </xs:sequence>
    <xs:attribute name="denominacion" type="xs:string" use="required"/>
    <xs:attribute name="categoria" type="xs:string" use="required"/>
 </xs:complexType>

 <xs:complexType name="tipoJugadores">
    <xs:sequence maxOccurs="unbounded">
       <xs:element ref="jugador"/>
    </xs:sequence>
 </xs:complexType>

 <xs:complexType name="tipoJugador">
    <xs:complexContent>
```

```
      <xs:extension base="datosIntegrantes">
        <xs:sequence>
          <xs:element ref="edad" minOccurs="0"/>
          <xs:element ref="foto"/>
        </xs:sequence>
        <xs:attribute ref="dorsal" use="required"/>
        <xs:attribute ref="posicion" use="optional"/>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoDorsal">
    <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[D][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
    <xs:restriction base="xs:string">
      <xs:enumeration value="base"/>
      <xs:enumeration value="escolta"/>
      <xs:enumeration value="alero"/>
      <xs:enumeration value="alapivot"/>
      <xs:enumeration value="pivot"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="tecnico"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoTecnico">
    <xs:complexContent>
      <xs:extension base="datosIntegrantes">
        <xs:attribute ref="codigo" use="required"/>
        <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
    <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[T][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

  <xs:complexType name="datosIntegrantes">
    <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:element ref="apellidos"/>
      <xs:element ref="nacionalidad" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="tipoEdad">
    <xs:restriction base="xs:positiveInteger">
```

```
        <xs:maxInclusive value="100"/>
    </xs:restriction>
 </xs:simpleType>

 <xs:complexType name="tipoFoto" mixed="true">
    <xs:simpleContent>
        <xs:extension base="xs:string">
            <xs:attribute ref="formato" use="required"/>
        </xs:extension>
    </xs:simpleContent>
 </xs:complexType>

 <xs:simpleType name="tipoFormatoImagen">
    <xs:restriction base="xs:string">
        <xs:enumeration value="GIF"/>
        <xs:enumeration value="JPG"/>
    </xs:restriction>
 </xs:simpleType>

</xs:schema>
```

2.11. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
     xs:noNamespaceSchemaLocation="equipo.xsd"
     denominacion="MiEquipo" categoria="Liga1”
     federacion="Federación de Baloncesto" flogo="LogotipoFederacion.gif">
  <jugadores>
      <jugador dorsal="D01" posicion="base">
          <nombre>Jugador1</nombre>
          <apellidos>Primero</apellidos>
          <nacionalidad>Estadounidense</nacionalidad>
          <foto formato="JPG">FotoD01.jpg</foto>
      </jugador>
      <jugador dorsal="D02">
          <nombre>Jugador2</nombre>
          <apellidos>Segundo</apellidos>
          <edad>21</edad>
          <foto formato="GIF">FotoD02.jpg</foto>
      </jugador>
  </jugadores>
  <tecnicos>
      <tecnico codigo="T01" cargo="entrenador">
          <nombre>Técnico1</nombre>
          <apellidos>Ape1</apellidos>
          <nacionalidad>Española</nacionalidad>
      </tecnico>
      <tecnico codigo="T02" cargo="fisio">
          <nombre>Técnico2</nombre>
          <apellidos>Ape2</apellidos>
      </tecnico>
  </tecnicos>
```

```
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Se le han añadido al elemento "equipo" (nodo raíz) los siguientes atributos:

- "federacion": atributo obligatorio, tipo texto. Se reflejará, de forma obligatoria, la denominación de la Federación (por lo tanto, será un valor fijo).
- "flogo": atributo obligatorio, tipo texto. Se reflejará, de forma obligatoria, la ruta al logotipo de la Federación (por lo tanto, será un valor fijo).

(Recordar que, a diferencia de los DTD, en los Esquemas XML no está permitido el uso de entidades (en el esquema se podría indicar que el tipo de dato es Entidad, pero no se validaría su contenido, que debería aparecer definido en el documento XML). Para solventarlo usaremos atributos tipo cadena de texto con valor fijado (fixed determina un valor fijo y por defecto) y lo marcaremos como opcional (si en el documento XML aparece tendrá que coincidir con el valor fijado. Si no aparece adoptará el valor fijo por defecto).
(No es necesario indicar que es opcional, dado que se toma por defecto, aunque lo indicamos para mejorar la comprensión).
(No se realiza una declaración independiente de los atributos dado que su contenido será de un tipo simple predefinido sin restricciones).
Agregar los atributos al "tipoEquipo":

```
<xs:attribute name="federacion" type="xs:string" fixed="Federación de Baloncesto" use="optional"/>
<xs:attribute name="flogo" type="xs:string" fixed="LogotipoFederacion.gif" use="optional"/>
```

▫ Esquema XML:

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

 <!-- Notaciones -->
 <xs:notation name="GIF" system="image/gif"/>
 <xs:notation name="JPG" system="image/jpeg"/>

 <!-- Elementos -->
 <xs:element name="equipo" type="tipoEquipo"/>

 <xs:element name="jugadores" type="tipoJugadores"/>
 <xs:element name="jugador" type="tipoJugador"/>

 <xs:element name="tecnicos" type="tipoTecnicos"/>
 <xs:element name="tecnico" type="tipoTecnico"/>

 <xs:element name="nombre" type="xs:string"/>
 <xs:element name="apellidos" type="xs:string"/>
 <xs:element name="nacionalidad" type="xs:string"/>
 <xs:element name="edad" type="tipoEdad"/>
 <xs:element name="foto" type="tipoFoto"/>

 <!-- Atributos -->
 <xs:attribute name="dorsal" type="tipoDorsal"/>
 <xs:attribute name="posicion" type="tipoPosicion"/>

 <xs:attribute name="codigo" type="tipoCodigoTecnico"/>
```

```
 <xs:attribute name="cargo" type="xs:NMTOKEN"/>

 <xs:attribute name="formato" type="tipoFormatoImagen"/>

 <!-- Tipos de Datos (construidos y complejos) -->
 <xs:complexType name="tipoEquipo">
    <xs:sequence>
       <xs:element ref="jugadores"/>
       <xs:element ref="tecnicos"/>
    </xs:sequence>
    <xs:attribute name="denominacion" type="xs:string" use="required"/>
    <xs:attribute name="categoria" type="xs:string" use="required"/>
    <xs:attribute name="federacion" type="xs:string" fixed="Federación de Baloncesto"
use="optional"/>
    <xs:attribute name="flogo" type="xs:string" fixed="LogotipoFederacion.gif"
use="optional"/>
 </xs:complexType>

 <xs:complexType name="tipoJugadores">
    <xs:sequence maxOccurs="unbounded">
       <xs:element ref="jugador"/>
    </xs:sequence>
 </xs:complexType>

 <xs:complexType name="tipoJugador">
    <xs:complexContent>
       <xs:extension base="datosIntegrantes">
          <xs:sequence>
             <xs:element ref="edad" minOccurs="0"/>
             <xs:element ref="foto"/>
          </xs:sequence>
          <xs:attribute ref="dorsal" use="required"/>
          <xs:attribute ref="posicion" use="optional"/>
       </xs:extension>
    </xs:complexContent>
 </xs:complexType>

 <xs:simpleType name="tipoDorsal">
    <xs:restriction base="xs:ID">
       <xs:length value="3"/>
       <xs:pattern value="[D][0-9][0-9]"/>
    </xs:restriction>
 </xs:simpleType>

 <xs:simpleType name="tipoPosicion">
    <xs:restriction base="xs:string">
       <xs:enumeration value="base"/>
       <xs:enumeration value="escolta"/>
       <xs:enumeration value="alero"/>
       <xs:enumeration value="alapivot"/>
       <xs:enumeration value="pivot"/>
    </xs:restriction>
 </xs:simpleType>

 <xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
       <xs:element ref="tecnico"/>
    </xs:sequence>
 </xs:complexType>
```

```
<xs:complexType name="tipoTecnico">
   <xs:complexContent>
      <xs:extension base="datosIntegrantes">
         <xs:attribute ref="codigo" use="required"/>
         <xs:attribute ref="cargo" use="required"/>
      </xs:extension>
   </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
   <xs:restriction base="xs:ID">
      <xs:length value="3"/>
      <xs:pattern value="[T][0-9][0-9]"/>
   </xs:restriction>
</xs:simpleType>

 <xs:complexType name="datosIntegrantes">
   <xs:sequence>
      <xs:element ref="nombre"/>
      <xs:element ref="apellidos"/>
      <xs:element ref="nacionalidad" minOccurs="0"/>
   </xs:sequence>
</xs:complexType>

<xs:simpleType name="tipoEdad">
   <xs:restriction base="xs:positiveInteger">
      <xs:maxInclusive value="100"/>
   </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoFoto" mixed="true">
   <xs:simpleContent>
      <xs:extension base="xs:string">
         <xs:attribute ref="formato" use="required"/>
      </xs:extension>
   </xs:simpleContent>
</xs:complexType>

<xs:simpleType name="tipoFormatoImagen">
   <xs:restriction base="xs:string">
      <xs:enumeration value="GIF"/>
      <xs:enumeration value="JPG"/>
   </xs:restriction>
</xs:simpleType>

</xs:schema>
```

2.12. Dado el siguiente documento XML, bien formado

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="equipo.xsd"
    denominacion="MiEquipo" categoria="Liga1”
    federacion="Federación de Baloncesto" flogo="LogotipoFederacion.gif">
 <jugadores>
      <jugador dorsal="D01" posicion="base">
         <nombre>Jugador1</nombre>
```

```
        <apellidos>Primero</apellidos>
        <nacionalidad>Estadounidense</nacionalidad>
        <foto formato="JPG">FotoD01.jpg</foto>
    </jugador>
    <jugador dorsal="D02">
        <nombre>Jugador2</nombre>
        <apellidos>Segundo</apellidos>
        <edad>21</edad>
        <foto formato="GIF">FotoD02.jpg</foto>
    </jugador>
</jugadores>
<tecnicos>
    <tecnico codigo="T01" cargo="entrenador">
        <nombre>Técnico1</nombre>
        <apellidos>Ape1</apellidos>
        <nacionalidad>Española</nacionalidad>
    </tecnico>
    <tecnico codigo="T02" cargo="fisio">
        <nombre>Técnico2</nombre>
        <apellidos>Ape2</apellidos>
    </tecnico>
</tecnicos>
<quinteto integra="D01 D02"/>
</equipo>
```

Construir su Esquema XML, en documento externo privado, y validar.
Se le han añadido al elemento "equipo" (nodo raíz) el elemento "quinteto", que presenta la siguiente composición:

- Es un elemento vacío, no contendrá información ni elementos hijo.
- Atributo "integra": atributo obligatorio, que hará referencia a los dorsales de los jugadores que integrarán el quinteto inicial en el partido.

Definición del elemento "quinteto":

```
<xs:element name="quinteto" type="tipoQuinteto"/>
```

Definición del "tipoQuinteto":

```
<xs:complexType name="tipoQuinteto">
    <xs:attribute name="integra" type="xs:IDREFS" use="required"/>
</xs:complexType>
```

Añadir el elemento "quinteto" al "tipoEquipo":

```
    <xs:sequence>
        <xs:element ref="jugadores"/>
        <xs:element ref="tecnicos"/>
        <xs:element ref="quinteto"/>
    </xs:sequence>
```

▫ Esquema XML:

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

```
<!-- Notaciones -->
<xs:notation name="GIF" system="image/gif"/>
<xs:notation name="JPG" system="image/jpeg"/>

<!-- Elementos -->
<xs:element name="equipo" type="tipoEquipo"/>

<xs:element name="jugadores" type="tipoJugadores"/>
<xs:element name="jugador" type="tipoJugador"/>

<xs:element name="tecnicos" type="tipoTecnicos"/>
<xs:element name="tecnico" type="tipoTecnico"/>

<xs:element name="nombre" type="xs:string"/>
<xs:element name="apellidos" type="xs:string"/>
<xs:element name="nacionalidad" type="xs:string"/>
<xs:element name="edad" type="tipoEdad"/>
<xs:element name="foto" type="tipoFoto"/>

<xs:element name="quinteto" type="tipoQuinteto"/>

<!-- Atributos -->
<xs:attribute name="dorsal" type="tipoDorsal"/>
<xs:attribute name="posicion" type="tipoPosicion"/>

<xs:attribute name="codigo" type="tipoCodigoTecnico"/>
<xs:attribute name="cargo" type="xs:NMTOKEN"/>

<xs:attribute name="formato" type="tipoFormatoImagen"/>

<!-- Tipos de Datos (construidos y complejos) -->
<xs:complexType name="tipoEquipo">
   <xs:sequence>
      <xs:element ref="jugadores"/>
      <xs:element ref="tecnicos"/>
      <xs:element ref="quinteto" minOccurs="0"/>
   </xs:sequence>
   <xs:attribute name="denominacion" type="xs:string" use="required"/>
   <xs:attribute name="categoria" type="xs:string" use="required"/>
   <xs:attribute name="federacion" type="xs:string" fixed="Federación de Baloncesto"
use="optional"/>
   <xs:attribute name="flogo" type="xs:string" fixed="LogotipoFederacion.gif"
use="optional"/>
</xs:complexType>

<xs:complexType name="tipoJugadores">
   <xs:sequence maxOccurs="unbounded">
      <xs:element ref="jugador"/>
   </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoJugador">
   <xs:complexContent>
      <xs:extension base="datosIntegrantes">
         <xs:sequence>
            <xs:element ref="edad" minOccurs="0"/>
            <xs:element ref="foto"/>
         </xs:sequence>
         <xs:attribute ref="dorsal" use="required"/>
         <xs:attribute ref="posicion" use="optional"/>
```

```
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoDorsal">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[D][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
    <xs:restriction base="xs:string">
        <xs:enumeration value="base"/>
        <xs:enumeration value="escolta"/>
        <xs:enumeration value="alero"/>
        <xs:enumeration value="alapivot"/>
        <xs:enumeration value="pivot"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="tecnico"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoTecnico">
    <xs:complexContent>
        <xs:extension base="datosIntegrantes">
            <xs:attribute ref="codigo" use="required"/>
            <xs:attribute ref="cargo" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[T][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

  <xs:complexType name="datosIntegrantes">
    <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="apellidos"/>
        <xs:element ref="nacionalidad" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>

<xs:simpleType name="tipoEdad">
    <xs:restriction base="xs:positiveInteger">
        <xs:maxInclusive value="100"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoFoto" mixed="true">
    <xs:simpleContent>
        <xs:extension base="xs:string">
```

```
            <xs:attribute ref="formato" use="required"/>
        </xs:extension>
    </xs:simpleContent>
 </xs:complexType>

 <xs:simpleType name="tipoFormatoImagen">
    <xs:restriction base="xs:string">
        <xs:enumeration value="GIF"/>
        <xs:enumeration value="JPG"/>
    </xs:restriction>
 </xs:simpleType>

 <xs:complexType name="tipoQuinteto">
    <xs:attribute name="integra" type="xs:IDREFS" use="required"/>
 </xs:complexType>

</xs:schema>
```

2.13. Elemento quinteto sea opcional.

Modificar el Esquema XML para incorporar la posibilidad que los documentos XML sean utilizados para enviar a la Federación la composición de la plantilla al inicio de la temporada (nodo raíz compuesto de jugadores y técnicos) y para enviar a la Federación el equipo en cada partido (nodo raíz compuesto de jugadores, técnicos y quinteto).

(Nota: en los Esquemas XML no es posible hacer uso de secciones condicionales, de uso en el DTD. Para ello podemos indicar que el elemento podrá no aparecer.)

Modificamos la declaración del "tipoEquipo":

```
<xs:element ref="quinteto" minOccurs="0"/>
```

- Documento XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<equipo xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
    xs:noNamespaceSchemaLocation="equipo.xsd"
    denominacion="MiEquipo" categoria="Liga1"
    federacion="Federación de Baloncesto" flogo="LogotipoFederacion.gif">
 <jugadores>
    <jugador dorsal="D01" posicion="base">
        <nombre>Jugador1</nombre>
        <apellidos>Primero</apellidos>
        <nacionalidad>Estadounidense</nacionalidad>
        <foto formato="JPG">FotoD01.jpg</foto>
    </jugador>
    <jugador dorsal="D02">
        <nombre>Jugador2</nombre>
        <apellidos>Segundo</apellidos>
        <edad>21</edad>
        <foto formato="GIF">FotoD02.jpg</foto>
    </jugador>
 </jugadores>
 <tecnicos>
    <tecnico codigo="T01" cargo="entrenador">
        <nombre>Técnico1</nombre>
        <apellidos>Ape1</apellidos>
        <nacionalidad>Española</nacionalidad>
    </tecnico>
    <tecnico codigo="T02" cargo="fisio">
```

```
        <nombre>Técnico2</nombre>
        <apellidos>Ape2</apellidos>
    </tecnico>
  </tecnicos>
  <quinteto integra="D01 D02"/>
</equipo>
```

- Esquema XML:

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

  <!-- Notaciones -->
  <xs:notation name="GIF" system="image/gif"/>
  <xs:notation name="JPG" system="image/jpeg"/>

  <!-- Elementos -->
  <xs:element name="equipo" type="tipoEquipo"/>

  <xs:element name="jugadores" type="tipoJugadores"/>
  <xs:element name="jugador" type="tipoJugador"/>

  <xs:element name="tecnicos" type="tipoTecnicos"/>
  <xs:element name="tecnico" type="tipoTecnico"/>

  <xs:element name="nombre" type="xs:string"/>
  <xs:element name="apellidos" type="xs:string"/>
  <xs:element name="nacionalidad" type="xs:string"/>
  <xs:element name="edad" type="tipoEdad"/>
  <xs:element name="foto" type="tipoFoto"/>

  <xs:element name="quinteto" type="tipoQuinteto"/>

  <!-- Atributos -->
  <xs:attribute name="dorsal" type="tipoDorsal"/>
  <xs:attribute name="posicion" type="tipoPosicion"/>

  <xs:attribute name="codigo" type="tipoCodigoTecnico"/>
  <xs:attribute name="cargo" type="xs:NMTOKEN"/>

  <xs:attribute name="formato" type="tipoFormatoImagen"/>

  <!-- Tipos de Datos (construidos y complejos) -->
  <xs:complexType name="tipoEquipo">
    <xs:sequence>
      <xs:element ref="jugadores"/>
      <xs:element ref="tecnicos"/>
      <xs:element ref="quinteto" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="denominacion" type="xs:string" use="required"/>
    <xs:attribute name="categoria" type="xs:string" use="required"/>
    <xs:attribute name="federacion" type="xs:string" fixed="Federación de Baloncesto"
use="optional"/>
    <xs:attribute name="flogo" type="xs:string" fixed="LogotipoFederacion.gif"
use="optional"/>
  </xs:complexType>

  <xs:complexType name="tipoJugadores">
    <xs:sequence maxOccurs="unbounded">
      <xs:element ref="jugador"/>
```

```
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoJugador">
    <xs:complexContent>
        <xs:extension base="datosIntegrantes">
            <xs:sequence>
                <xs:element ref="edad" minOccurs="0"/>
                <xs:element ref="foto"/>
            </xs:sequence>
            <xs:attribute ref="dorsal" use="required"/>
            <xs:attribute ref="posicion" use="optional"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoDorsal">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[D][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

<xs:simpleType name="tipoPosicion">
    <xs:restriction base="xs:string">
        <xs:enumeration value="base"/>
        <xs:enumeration value="escolta"/>
        <xs:enumeration value="alero"/>
        <xs:enumeration value="alapivot"/>
        <xs:enumeration value="pivot"/>
    </xs:restriction>
</xs:simpleType>

<xs:complexType name="tipoTecnicos">
    <xs:sequence maxOccurs="unbounded">
        <xs:element ref="tecnico"/>
    </xs:sequence>
</xs:complexType>

<xs:complexType name="tipoTecnico">
    <xs:complexContent>
        <xs:extension base="datosIntegrantes">
            <xs:attribute ref="codigo" use="required"/>
            <xs:attribute ref="cargo" use="required"/>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

<xs:simpleType name="tipoCodigoTecnico">
    <xs:restriction base="xs:ID">
        <xs:length value="3"/>
        <xs:pattern value="[T][0-9][0-9]"/>
    </xs:restriction>
</xs:simpleType>

  <xs:complexType name="datosIntegrantes">
    <xs:sequence>
        <xs:element ref="nombre"/>
        <xs:element ref="apellidos"/>
        <xs:element ref="nacionalidad" minOccurs="0"/>
```

```
    </xs:sequence>
  </xs:complexType>

  <xs:simpleType name="tipoEdad">
    <xs:restriction base="xs:positiveInteger">
      <xs:maxInclusive value="100"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tipoFoto" mixed="true">
    <xs:simpleContent>
      <xs:extension base="xs:string">
        <xs:attribute ref="formato" use="required"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>

  <xs:simpleType name="tipoFormatoImagen">
    <xs:restriction base="xs:string">
      <xs:enumeration value="GIF"/>
      <xs:enumeration value="JPG"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="tipoQuinteto">
    <xs:attribute name="integra" type="xs:IDREFS" use="required"/>
  </xs:complexType>

</xs:schema>
```

3. Ejercicio “PERSONA”

Define el Esquema XML (persona.xsd) que permita validar el siguiente documento XML y añade la correspondiente asociación al esquema de validación:

▫ Documento XML (persona.xml)

```
<?xml version="1.0" encoding="UTF-8"?>
<persona dni="12.345.678L" estadoCivil="casado">
    <nombre>María Pilar</nombre>
    <apellido>Sánchez</apellido>
    <edad>25</edad>
    <enActivo/>
</persona>
```

Restricciones:

- Una persona puede estar en activo o no estarlo (interpretamos que aparecerá la etiqueta “enActivo” o la etiqueta “noActivo”).
- La edad, lógicamente, tiene que ser un entero mayor a cero.
- El DNI es un identificador obligatorio. Su formato será xx.xxx.xxxL.
- El estado civil puede ser soltero, casado o divorciado, y por defecto es soltero.

Valida el documento XML del ejercicio anterior utilizando "XML Copy Editor". ¿Se produce algún error? ¿A qué se debe? ¿Cómo podrías corregirlo?

4. Ejercicio “TICKETS”

Completa el Esquema XML del siguiente documento XML.

▫ Documento XML: “tickets.xml”.

```
<?xml version="1.0" encoding="UTF-8"?>
<listatickets>
    <ticket>
        <numero>0001</numero>
        <fecha>01/02/2016</fecha>
        <hora>22:00</hora>
        <producto>
            <nombre>Agua</nombre>
            <precio moneda="euro">1.00</precio>
        </producto>
        <producto>
            <nombre>Refresco</nombre>
            <precio moneda="euro">1.50</precio>
        </producto>
        <total moneda="euro">2.50</total>
    </ticket>
    <ticket>
        <numero>0002</numero>
        <fecha>01/02/2016</fecha>
        <hora>22:05</hora>
        <producto>
            <nombre>Refresco</nombre>
            <precio moneda="euro">1.50</precio>
        </producto>
        <producto>
            <nombre>Refresco</nombre>
            <precio moneda="euro">1.50</precio>
        </producto>
        <total moneda="euro">3.00</total>
    </ticket>
</listatickets>
```

El atributo “moneda” adoptará el valor “euro” por defecto y tipo de dato “NMTOKEN”.

▫ Esquema XML: “tickets.xsd”.

5. Ejercicio “ALUMNO”

Define en un esquema XSD (“alumno.xsd”) los elementos necesarios para almacenar la ficha de un alumno compuesta por: código de alumno (identificador único, atributo de alumno, formato Axx), nombre, primer apellido, segundo apellido (puede no aparecer), DNI (con formato xx.xxx.xxx-L) o NIE (con formato [X o Y]x.xxx.xxx-L), fecha de nacimiento (formato año-mes-día), curso (formato nivel con tres letras y nº, por ejemplo, ESO1) y si ha pagado la cuota (“cuota” puede adoptar los valores “pagada” o ”no pagada”, por defecto “no pagada”) (atributo de alumno).

Escribe un ejemplo de documento XML (“alumno.xml”).

▫ Documento XML: “alumno.xml”

▫ Esquema XML: “alumno.xsd”

6. Ejercicio “HELADERÍA”

Diseña el esquema XML para el siguiente documento XML y las consideraciones posteriores.

▫ Documento XML: “heladeria.xml”

```
<?xml version="1.0" encoding="UTF-8"?>
<heladeria xmlns:xs="http://www.w3.org/2001/XMLSchema-instance"
           xs:noNamespaceSchemaLocation="heladeria.xsd">
    <helado fabricación="2015-01-30">
        <chocolate>250</chocolate>
        <turron>300</turron>
    </helado>
    <helado fabricación="2015-01-30">
        <fresa>500</fresa>
        <vainilla>200</vainilla>
    </helado>
</heladeria>
```

- Como primer elemento de helado podemos elegir chocolate o fresa (pero no los dos al mismo tiempo). Estos elementos almacenarán la cantidad en gramos (número entero entre 0 y 1000).
- Como segundo elemento de helado se podrá elegir entre vainilla, turrón y nata. Estos elementos almacenarán la cantidad en gramos (número entero entre 0 y 1000).
- El atributo fabricación indicará la fecha en la que se fabricó el mismo mediante el tipo de dato fecha.

▫ Esquema XML: “heladeria.xsd”