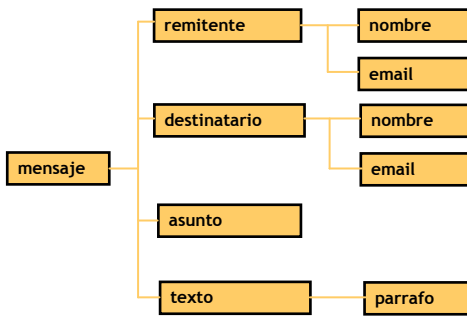


- Partiendo del siguiente diagrama, donde los nodos representan elementos, definir un esquema correspondiente a dicha estructura XML y crear un documento XML que sea válido.



- De acuerdo al siguiente documento XML de películas,
  - Crear un esquema que se corresponda con la estructura del documento.
  - Validar el documento XML con el esquema creado.

```

<?xml version="1.0"?>
<catalogoPelículas>
  <película>
    <titulo>The Matrix</titulo>
    <duracion>136</duracion>
    <genero>Sci-Fi</genero>
    <actores>
      <actor>Keanu Reeves</actor>
      <actor>Laurence Fishburne</actor>
      <actor>Carrie Ann Moss</actor>
    </actores>
    <fecha>1999</fecha>
    <director>Wachowski Brothers</director>
    <formato>DVD</formato>
    <trailer>matrix.avi</trailer>
  </película>
  <película>
    <titulo>Titanic</titulo>
    <duracion>194</duracion>
    <genero>Drama</genero>
    <actores>
      <actor>Leonardo DiCaprio</actor>
      <actor>Kate Winslet</actor>
    </actores>
    <fecha>1999</fecha>
    <formato>DVD</formato>
    <cartel>titanic.jpg</cartel>
  </película>
  <película>
    <titulo>The Sixth Sense</titulo>
    <duracion>106</duracion>
    <genero>Thriller</genero>
    <actores>
      <actor>Bruce Willis</actor>
      <actor>Haley Joel Osment</actor>
    </actores>
    <fecha></fecha>
    <director>M. Night Shyamalan</director>
    <formato>HD720</formato>
    <trailer>sixthSense.avi</trailer>
  </película>
</catalogoPelículas>
  
```

- Modificar el esquema del ejercicio 1º añadiendo una restricción para el contenido de los *email* usando la siguiente plantilla:

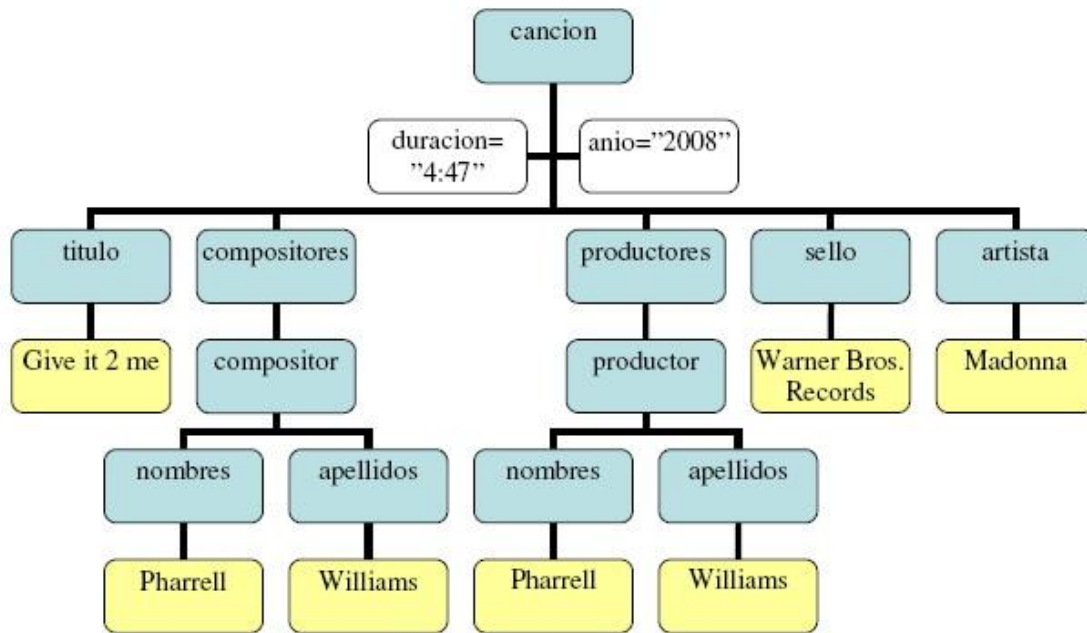
atext+(\. atext+)\*@atext+(\. atext+)\*

sustituyendo *atext* por el conjunto de caracteres permitidos en una dirección de correo (letras mayúsculas y minúsculas, dígitos y los caracteres “-” y “\_”).

4. Modificar el esquema del ejercicio 2º añadiendo las siguientes restricciones:
- El género al que pertenece una película solo podrá tener uno de los siguientes valores: *drama*, *comedia*, *terror*, *thriller* o *sci-fi*.
  - La fecha de la película será un entero no inferior a 1900.
  - El formato de la película será DVD o VHS, siendo DVD el valor por defecto.
  - Se podrán indicar como máximo 3 actores por película.

Validar el documento XML con el nuevo esquema.

5. Partiendo del siguiente diagrama, donde los nodos en azul representan elementos, los blancos atributos y los amarillos contenidos, definir un esquema válido para dicha estructura y crear un documento XML válido. Definir para el elemento *cancion* un tipo global (*T\_cancion*) y para la pareja de elementos *nombre* y *apellidos* otro tipo de nombre (*T\_persona*)



6. Ampliar la jerarquía de datos propuesta del ejercicio anterior para representar la estructura de un CD completo. Para ello en el documento XML crear un nuevo elemento CD que contenga:
- Un atributo *titulo* que tome como valor el título del álbum.
  - Dos elementos:
    - Un elemento *canciones* que a su vez puede contener varios elementos *cancion*.
    - Un elemento *fecha* con tres atributos: *dia*, *mes* y *anio* para indicar la fecha de salida al mercado del CD.
  - Crear un esquema que valide al nuevo documento XML.
7. Modificar el esquema del ejercicio anterior de forma que el elemento *cancion* tendrá dos nuevos atributos. El primero obligatorio denominado *precio* con los posibles valores 0.99, 1.49 y 1.99; y el segundo llamado *moneda* con dos posibles valores *dólar* o *euro*, siendo este último el valor por defecto. Crear un documento XML válido frente a este nuevo esquema.
8. Crear un esquema correspondiente al documento XML y luego validarlo teniendo en cuenta:
- Después de uno o más elemento *capitulo* podrá haber un elemento *separacion*.
  - El atributo *materia* sólo podrá tomar los valores *XML* (valor por defecto) o *Java*
  - El atributo *dificil* sólo podrá tomar los valores *no* (valor por defecto) o *si*

```
<?xml version="1.0" encoding="UTF-8"?>
<libro xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="libro.xsd">
  <titulo>Java y XML</titulo>
  <contenido>
    <capitulo materia="XML">
      <tema>Introducción</tema>
      <seccion apartados="7">Qué es</seccion>
      <seccion apartados="3">Cómo se usa</seccion>
    </capitulo>
  </contenido>
</libro>
```

```

<capitulo materia="XML">
  <tema>Creando XML</tema>
  <seccion apartados="0">Un documento XML</seccion>
  <seccion apartados="2">La cabecera</seccion>
  <seccion apartados="6">El contenido</seccion>
</capitulo>
<capitulo>
  <tema>Analizando XML</tema>
  <seccion apartados="3">Preparación</seccion>
  <seccion apartados="3" dificil="si">SAX</seccion>
  <seccion apartados="9" dificil="si">Manejadores</seccion>
  <seccion apartados="0">Una forma mejor de cargar el analizador</seccion>
</capitulo>
<separacion/>
<capitulo materia="Java">
  <tema>JDOM</tema>
  <seccion apartados="2">Introducción</seccion>
  <seccion apartados="4" dificil="si">DOM&JDOM</seccion>
</capitulo>
</contenido>
<copyright>2000</copyright>
</libro>

```

9. Crear un esquema correspondiente al documento XML posterior y validarlo.

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<FAQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="faq.xsd">
  <INFO>
    <TITULO>WINDOWS98? SOLO SI ME OBLIGAN!</TITULO>
    <AUTOR>Fernando Lorenzo García</AUTOR>
    <EMAIL>Fer@yahoo.es</EMAIL>
    <VERSION>1.0</VERSION>
    <FECHA>1999-05-20</FECHA>
  </INFO>
  <PART NO="P1" TITLE="W95">
    <Q N="1">
      <QTEXT>¿Por qué tengo que instalar W95?</QTEXT>
      <A>A mí también me parece una buena pregunta.</A>
      <A>Y a mí ...</A>
    </Q>
    <Q N="2">
      <QTEXT>¿Y de que me vale?</QTEXT>
      <A>Otra ingeniosa pregunta.</A>
    </Q>
  </PART>
  <PART NO="P2" TITLE="DOS">
    <Q N="1">
      <QTEXT>¿Puedo mantener DOS y W95?</QTEXT>
      <A>Pues sí.</A>
    </Q>
    <Q N="2">
      <QTEXT>¿Tengo que cambiar de ordenador?</QTEXT>
      <A>Depende....</A>
    </Q>
    <Q N="3">
      <QTEXT>¿De qué depende?</QTEXT>
    </Q>
  </PART>
</FAQ>

```

10. Dado el siguiente DTD que define un lenguaje de marcado para almacenar datos de libros en una librería virtual, obtener su esquema equivalente

```

<!ELEMENT biblioteca (libro)+>
<!ELEMENT libro ( titulo, disponible, autor, formato, publicacion, precio, descuento, enlancelibro )>
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT disponible EMPTY>
<!ATTLIST disponible tiempo NMTOKEN #REQUIRED>
<!ATTLIST disponible unidad NMTOKEN #REQUIRED>
<!ELEMENT autor (#PCDATA)>
<!ELEMENT formato (#PCDATA)>
<!ELEMENT publicacion (#PCDATA)>
<!ELEMENT precio EMPTY>

```

```

<!ATTLIST precio cantidad CDATA #REQUIRED>
<!ATTLIST precio moneda (euro | dolar) #REQUIRED>
<!ELEMENT descuento EMPTY>
<!ATTLIST descuento cantidad CDATA #IMPLIED>
<!ELEMENT enlancelibro EMPTY>
<!ATTLIST enlancelibro href CDATA #REQUIRED>
<!ELEMENT portada EMPTY>
<!ATTLIST portada archivo CDATA #IMPLIED>

```

11. Diseñar un esquema para crear un documento para registrar la información de un préstamo.
- En el préstamo se indicará:
    - El dni, nombre y apellidos del bibliotecario.
    - Fecha del préstamo y de devolución.
    - Datos del lector (dni, nombre, apellidos, teléfono y dirección).
    - La dirección tendrá: un tipo de calle (que puede ser calle o avenida), nombre calle, número, piso y letra, código postal [1001-52080], localidad y provincia.
    - Un máximo de tres ejemplares en préstamo. Para cada uno de ellos: el número de registro, título, autor(es).
    - El préstamo tendrá un atributo numérico que servirá como identificador.
  - Crear un documento según el esquema definido y verificar que sea válido.
12. Dado el siguiente DTD que define un lenguaje de marcado para almacenar datos de personas con direcciones y e-mail, obtener su esquema equivalente.

```

<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT listin (persona)+>
<!ELEMENT persona (nombre, email*, relacion?)>
<!ATTLIST persona id ID #REQUIRED>
<!ATTLIST persona sexo (hombre | mujer) #REQUIRED>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT relacion EMPTY>
<!ATTLIST relacion amigo-de IDREFS #IMPLIED enemigo-de IDREFS #IMPLIED>

```

13. A partir del siguiente DTD que guarda información de películas, obtener su esquema equivalente.

```

<!ELEMENT lista_peliculas (pelicula*)>
<!ELEMENT pelicula (titulo, director+)>
<!ATTLIST pelicula identificador ID #REQUIRED clasificacion (A | B | C | D | E) "C">
<!ELEMENT titulo (#PCDATA)>
<!ELEMENT director (nombre, inicial?, apellido)>
<!ELEMENT nombre (#PCDATA)>
<!ELEMENT inicial (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>

```

14. Definir un Schema que defina un lenguaje de marcado para guardar información de un conjunto de sorteos. Para cada sorteo se deberá registrar la fecha del sorteo y una lista de 6 números enteros comprendidos entre 1 y 49 con los números agraciados.
15. Escribir un documento XML que sea una instancia del Schema *facturas.xsd*.

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <!-- ===== -->
  <!-- Definiciones de tipos simples -->
  <!-- ===== -->

  <!-- ===== Tipo "tipoID" ===== -->
  <xsd:simpleType name="tipoID">
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:totalDigits value="4"/>
    </xsd:restriction>
  </xsd:simpleType>

  <!-- ===== Tipo "tipoIVA" ===== -->
  <xsd:simpleType name="tipoIVA">
    <xsd:restriction base="xsd:positiveInteger">
      <xsd:totalDigits value="2"/>
    </xsd:restriction>
  </xsd:simpleType>

  <!-- ===== Tipo "tipoReferencia" ===== -->

```

```

<xsd:simpleType name="tipoReferencia">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="5"/>
    <xsd:maxLength value="15"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Tipo "tipoDescripcion" ===== -->
<xsd:simpleType name="tipoDescripcion">
  <xsd:restriction base="xsd:string">
    <xsd:minLength value="1"/>
    <xsd:maxLength value="100"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Tipo "tipoPrecio" ===== -->
<xsd:simpleType name="tipoPrecio">
  <xsd:restriction base="xsd:decimal">
    <xsd:totalDigits value="7"/>
    <xsd:fractionDigits value="2"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Tipo "tipoReal2Decimales" ===== -->
<xsd:simpleType name="tipoReal2Decimales">
  <xsd:restriction base="xsd:decimal">
    <xsd:fractionDigits value="2"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Tipo "tipoCodPostal" ===== -->
<xsd:simpleType name="tipoCodPostal">
  <xsd:restriction base="xsd:positiveInteger">
    <xsd:totalDigits value="5"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Tipo "tipoFormaPago" ===== -->
<xsd:simpleType name="tipoFormaPago">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="efectivo"/>
    <xsd:enumeration value="tarjeta"/>
    <xsd:enumeration value="plazox"/>
  </xsd:restriction>
</xsd:simpleType>

<!-- ===== Definiciones de elementos de tipos simples -->
<!-- ===== -->

<!-- ===== Elementos de tipo string ===== -->
<xsd:element name="nombre" type="xsd:string"/>
<xsd:element name="dir" type="xsd:string"/>
<xsd:element name="provincia" type="xsd:string"/>
<xsd:element name="cif" type="xsd:string"/>
<xsd:element name="dir_env" type="xsd:string"/>

<!-- ===== Elementos de tipo numérico ===== -->
<xsd:element name="base" type="tipoReal2Decimales"/>
<xsd:element name="cuota_iva" type="tipoReal2Decimales"/>
<xsd:element name="total" type="tipoReal2Decimales"/>
<xsd:element name="ref" type="tipoReferencia"/>
<xsd:element name="desc" type="tipoDescripcion"/>
<xsd:element name="cant" type="xsd:positiveInteger"/>
<xsd:element name="precio" type="tipoPrecio"/>
<xsd:element name="importe" type="tipoReal2Decimales"/>

<!-- ===== Otros elementos de tipo simple ===== -->
<xsd:element name="fecha" type="xsd:date"/>

<!-- ===== -->

```

```

<!-- Definiciones de elementos de tipo complejo -->
<!-- ===== -->

<!-- ===== Elemento "telefono" ===== -->
<xsd:element name="telefono">
  <xsd:complexType>
    <xsd:attribute name="num_tel" type="xsd:string" fixed="917776688"/>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "fax" ===== -->
<xsd:element name="fax">
  <xsd:complexType>
    <xsd:attribute name="num_fax" type="xsd:string" use="required"
      fixed="917776699"/>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "poblacion" ===== -->
<xsd:element name="poblacion">
  <xsd:complexType>
    <xsd:simpleContent>
      <xsd:extension base="xsd:string">
        <xsd:attribute name="cod_postal" type="tipoCodPostal"/>
      </xsd:extension>
    </xsd:simpleContent>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "linea" ===== -->
<xsd:element name="linea">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="ref"/>
      <xsd:element ref="desc"/>
      <xsd:element ref="cant"/>
      <xsd:element ref="precio"/>
      <xsd:element ref="importe"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "datos_empresa" ===== -->
<xsd:element name="datos_empresa">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="nombre"/>
      <xsd:element ref="dir"/>
      <xsd:element ref="poblacion"/>
      <xsd:element ref="provincia"/>
      <xsd:element ref="cif"/>
      <xsd:element ref="telefono" minOccurs="0" maxOccurs="1"/>
      <xsd:element ref="fax" minOccurs="0" maxOccurs="1"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "datos_cliente" ===== -->
<xsd:element name="datos_cliente">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="nombre"/>
      <xsd:element ref="dir_env"/>
      <xsd:element ref="poblacion"/>
      <xsd:element ref="provincia"/>
    </xsd:sequence>
    <xsd:attribute name="n_cli" type="tipoID"/>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "datos_factura" ===== -->

```

```

<xsd:element name="datos_factura">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="fecha"/>
      <xsd:element ref="linea" minOccurs="0" maxOccurs="unbounded"/>
      <xsd:element ref="base"/>
      <xsd:element ref="cuota_iva"/>
      <xsd:element ref="total"/>
    </xsd:sequence>
    <xsd:attribute name="n_ped" type="tipoID"/>
    <xsd:attribute name="iva" type="tipoIVA"/>
    <xsd:attribute name="f_pago" type="tipoFormaPago" use="required"/>
    <xsd:attribute name="moneda" type="xsd:string" fixed="euro"/>
  </xsd:complexType>
</xsd:element>

<!-- ===== Elemento "factura" ===== -->
<xsd:element name="factura">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element ref="datos_empresa"/>
      <xsd:element ref="datos_cliente"/>
      <xsd:element ref="datos_factura"/>
    </xsd:sequence>
    <xsd:attribute name="n_fac" type="tipoID"/>
  </xsd:complexType>
</xsd:element>

</xsd:schema>

```

16. Definir un Schema que defina un lenguaje de marcado para guardar información de un conjunto de alumnos. Para cada alumno se deberá registrar su nombre y la notas obtenidas en cada una de las asignaturas matriculadas. El alumno podrá estar matriculado como máximo en 5 asignaturas. La nota se podrá almacenar como un entero entre 1 y 10, o como una cadena de caracteres. Las posibles cadenas son: "MD", "IN", "SF", "BI", NT y "SB". Terminado el Schema, construir un archivo XML que sea válido frente al Schema.
17. Un grupo de amigos (Antonio, Bartolo, Carmen, Daniel, Emilia, Felipe, Gerardo, Horacio y Fuencisla) han ideado un juego y necesitan que cada una de las cabezas de serie: Antonio, Emilia y Gerardo elijan entre 2 y 4 participantes en sus equipos. Las peticiones has quedado de la forma:

```

<?xml version="1.0" encoding="UTF-8"?>
<grupos xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="grupos.xsd">
  <grupo de="Antonio">Carmen Emilia Horacio</grupo>
  <grupo de="Emilia">Antonio Felipe</grupo>
  <grupo de="Gerardo">Antonio Daniel Emilia Felipe</grupo>
</grupos>

```

Se pide realizar un Schema que valide el documento anterior.