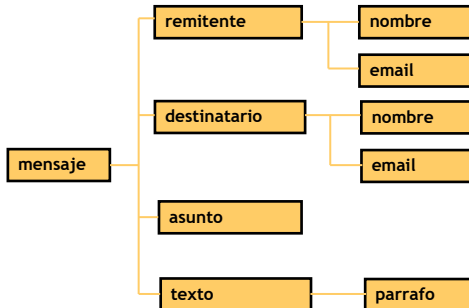


- Partiendo del siguiente diagrama, donde los nodos representan elementos, definir un esquema correspondiente a dicha estructura XML y crear un documento XML que sea válido. Posteriormente añadir al esquema una restricción para el contenido de los *email* usando la siguiente plantilla: `text+(\.text+)*@text+(\.text+)*` donde *text* representa el conjunto de caracteres permitidos: letras mayúsculas y minúsculas, dígitos y los caracteres “-” y “_”.



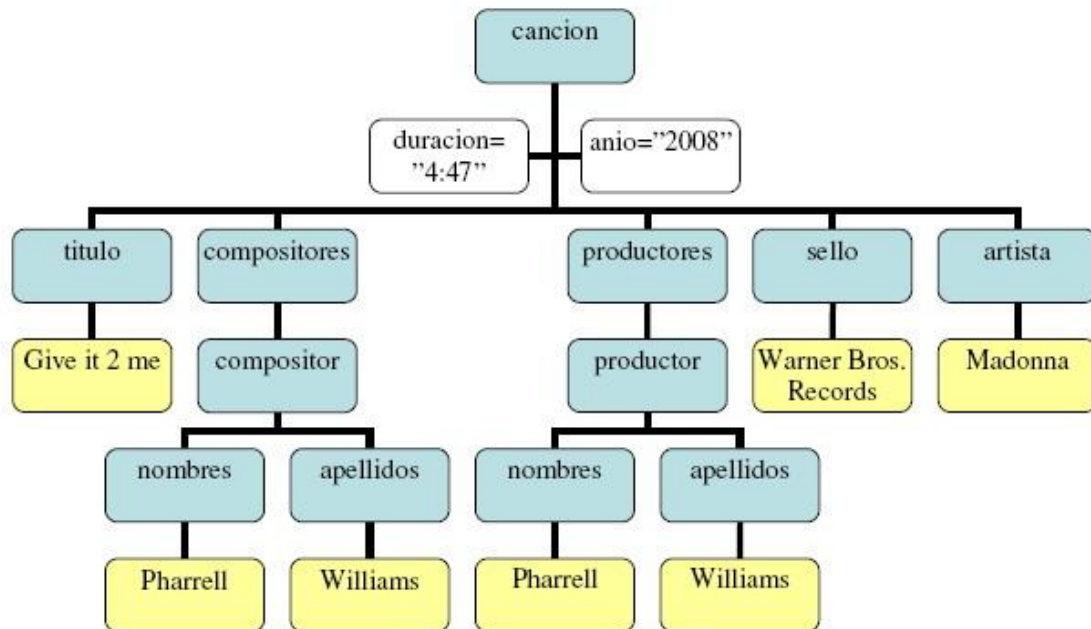
- De acuerdo al siguiente documento XML de películas,
 - Crear un esquema que se corresponda con la estructura del documento teniendo en cuenta:
 - El género al que pertenece una película solo podrá tener uno de los valores “Drama”, “Comedia”, “Terror”, “Thriller” o “Sci-Fi and Fantasy”.
 - La fecha de la película será un entero no inferior a 1900.
 - Los posibles formatos de película serán AVI, MOV, MP4 y FLV, siendo AVI el valor por defecto.
 - Se podrán indicar como máximo 3 actores por película, que incluirán elementos actor y/o actriz.
 - Validar XML con el esquema creado.

```

<?xml version="1.0"?>
<CatalogoPeliculas>
  <Pelicula>
    <Titulo>The Matrix</Titulo>
    <Duracion>136</Duracion>
    <Genero>Sci-Fi and Fantasy</Genero>
    <Actores>
      <Actor>Keanu Reeves</Actor>
      <Actor>Laurence Fishburne</Actor>
      <Actriz>Carrie Ann Moss</Actriz>
    </Actores>
    <Fecha>1999</Fecha>
    <Director>Wachowski Brothers</Director>
    <Formato>MP4</Formato>
  </Pelicula>
  <Pelicula>
    <Titulo>Titanic</Titulo>
    <Duracion>194</Duracion>
    <Genero>Drama</Genero>
    <Actores>
      <Actor>Leonardo DiCaprio</Actor>
      <Actriz>Kate Winslet</Actriz>
    </Actores>
    <Fecha>1999</Fecha>
    <Director>James Cameron</Director>
    <Formato>AVI</Formato>
  </Pelicula>
  <Pelicula>
    <Titulo>The Sixth Sense</Titulo>
    <Duracion>106</Duracion>
    <Genero>Thriller</Genero>
    <Actores>
      <Actor>Bruce Willis</Actor>
      <Actor>Haley Joel Osment</Actor>
    </Actores>
    <Fecha>1999</Fecha>
    <Director>M. Night Shyamalan</Director>
    <Formato>MP4</Formato>
  </Pelicula>
</CatalogoPeliculas>
  
```

```
</Pelicula>
</CatalogoPeliculas>
```

3. 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.



4. Ampliar la jerarquía de datos propuesta del ejercicio anterior para representar la estructura de un CD completo. Para ello crear un nuevo elemento “CD” que contenga:
- Un atributo “título” que tome como valor el título genérico del “CD”.
 - Dos elementos:
 - Un elemento “fecha” que almacena la fecha de salida al mercado del “CD”.
 - Un elemento “canciones” que a su vez puede contener varios elementos “cancion”.
 - Agregar los cambios producidos en la representación de la estructura, en un nuevo esquema.
5. Crear un esquema correspondiente al documento XML siguiente y luego validar éste.

```
<?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>Introduccion</tema>
      <seccion apartados="7">Qué es</seccion>
      <seccion apartados="3">Cómo se usa</seccion>
    </capitulo>
    <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>
  </contenido>
</libro>
```

```
<capitulo materia="Java">
  <tema>JDOM</tema>
  <seccion apartados="2">Introducción</seccion>
  <seccion apartados="4" dificil="si">DOM&amp;JDOM</seccion>
</capitulo>
</contenido>
<copyright>2000</copyright>
</libro>
```

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

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<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 Garcia</AUTOR>
    <EMAIL>Fer@yahoo.es</EMAIL>
    <VERSION>28.56</VERSION>
    <FECHA>1999-05-20</FECHA>
  </INFO>
  <PART NO="1" TITLE="W95">
    <Q NO="1">
      <QTEXT>¿Por qué tengo que instalar W95?</QTEXT>
      <A> A mi también me parece una buena pregunta.</A>
    </Q>
    <Q NO="2">
      <QTEXT>¿Y de que me vale?</QTEXT>
      <A>Otra ingeniosa pregunta.</A>
    </Q>
  </PART>
  <PART NO="2" TITLE="DOS">
    <Q NO="1">
      <QTEXT>¿Puedo mantener DOS y W95?</QTEXT>
      <A>Pues si.</A>
    </Q>
    <Q NO="2">
      <QTEXT>¿Tengo que cambiar de ordenador?</QTEXT>
      <A>Depende.</A>
    </Q>
  </PART>
</FAQ>
```

7. Crear un esquema y un documento XML que se valide con él para guardar la siguiente información sobre de los estudiantes de una clase:

- “nombre”
- “notas”, formado por seis valores escogidos entre éstos:
 - Un número entero comprendido entre 1 y 9.
 - NE o APRO (No evaluado y Aprobado, respectivamente).

8. Diseñar un esquema para crear documentos que registren la información de un préstamo.

- a) En el préstamo se indicará:
 1. DNI, nombre y apellidos del bibliotecario.
 2. Fecha del préstamo y de devolución.
 3. Datos del lector (DNI, nombre, apellidos, teléfono y dirección).
 4. La dirección tendrá: un tipo de vía (que puede, al menos, ser calle, avenida o plaza), nombre de la vía, número, piso y letra, código postal [1001-52080], localidad y provincia.
 5. Un máximo de tres ejemplares en préstamo. Para cada uno de ellos: el número de registro, título, autor(es).
 6. El préstamo tendrá un atributo numérico que servirá como identificador.
- b) Crear un documento según el esquema definido y verificar que sea válido.

9. Dado el siguiente DTD (Document Type Definition) 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 CDATA #REQUIRED >
<!ATTLIST disponible unidad CDATA #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 >

```

10. Dada la siguiente DTD que define un lenguaje de marcado para almacenar datos de personas con direcciones y e-mail, obtener su esquema equivalente.

```

<!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>

```

11. A partir de la 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)>

```