Relación 2 DTD

DTD (*Document Type Definition)* : mecanismo para expresar las reglas sobre lo que se va a permitir y lo que no en archivos XML.

1. Este documento guarda información sobre países. Comprueba que el documento está bien formado y es válido

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paises [
   <!ELEMENT paises (pais, ciudad?)*>
   <!ELEMENT pais (#PCDATA)>
   <!ELEMENT ciudad (#PCDATA)>
]>
<paises>
   <pais>italia</pais>
   <ciudad>florencia</ciudad>
   <pais>portugal</pais>
   <ciudad>lisboa</ciudad>
   <pais>francia</pais>
</paises>
```

2. El siguiente documento guarda información sobre periféricos. Analízalo y comprueba que está bien formado, pero no es válido. Propón los cambios necesarios para que sea válido.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE perifericos [
   <!ELEMENT perifericos (#PCDATA)>
]>

<perifericos>
   <periferico>impresora</periferico>
   <periferico>monitor</periferico>
   <periferico>teclado</periferico>
</perifericos>
```

3. Corrija los errores del siguiente documento DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE animales [
  <!ELEMENT animales (animal*)>
  <!ELEMENT animal (#PCDATA)>
]>

<animales>
  <perro>Caniche</perro>
  <gato>Siamés</gato>
</animales>
```

4. Unos programadores desean realizar una aplicación de pedidos teniendo en cuenta los siguientes requisitos:

   - La lista de pedidos será de 0 o más pedidos
   - Cada pedido tiene un número de referencia, una cantidad, un precio y un peso que puede ser opcional.

   Se pide:

   a) Crear un XML de ejemplo y la DTD asociada válidos
   b) Otro ejemplo no válido

5. Se desea crear un ejemplo de XML con una DTD asociada para ficheros de datos , con información de los productos de una empresa, en los que se ha decidido contemplar lo siguiente:

   - El fichero debe llevar una raíz <productos>
   - Dentro debe haber uno o más elementos <producto>
   - Los productos pueden ser alguno de los siguientes: <raton> , <teclado> o <monitor>
   - Todo ratón, teclado o monitor tiene siempre un código.
   - Todo ratón, teclado o monitor puede llevar un nombre.
   - Todo ratón, teclado o monitor puede llevar una descripción.

6. El departamento de ventas del comercio “MisMejoresCompras” ha decidido la siguiente estructura para ficheros de datos que se tengan que mover de un software a otro.

   - La raíz debe ser el elemento <listacompras>
   - Dentro de <listacompras> debe haber uno o más elementos <venta>
   - Una venta puede ser de uno de los dos tipos siguientes: <ventaacredito> o <ventainmediata>
   - Un elemento <ventaacredito> consta de un elemento <fechafinpago> que es optativo y un elemento <cantidad> que es obligatorio.
   - Un elemento <ventainmediata> consta de un elemento <cantidad> que es obligatorio y un elemento <divisa> que también es obligatorio.

   Se pide un ejemplo XML y un DTD asociado.

7. Una papelería desea tener un formato de almacenamiento de datos para reflejar la información de su inventario, teniendo en cuenta los siguientes requisitos:

   - El elemento raíz debe ser <inventario>
   - Dentro de inventario pueden ir elementos <lapiz>, <cuaderno> o <boligrafo> repetidos y en cualquier orden.
   - Todo <lapiz> puede tener una <dureza>
   - Todo cuaderno debe tener: <numhojas> y <estilo>
   - Todo boligrafo tiene un <precio> y puede o no tener especificado un <color>

   Se pide un ejemplo XML y el DTD asociada.

8. Unos programadores necesitan estructurar la información que intercambiarán los ficheros de sus aplicaciones para lo cual han determinado los siguientes requisitos.

   - Los ficheros deben tener un elemento <listafacturas>
   - Dentro de la lista debe haber una o más facturas.
   - Las facturas tienen un atributo fecha que es optativo.
   - Toda factura tiene un emisor, que es un elemento obligatorio, con el atributo CIF (obligatorio).
   - Todo emisor tiene un nombre, que es obligatorio, y puede tener o no un <volumenventas>.
   - Toda factura debe tener un elemento pagador, el cual tiene exactamente la misma estructura que emisor.
   - Toda factura tiene un elemento importe.

Escribe un posible documento XML y la DTD que refleje los requisitos del enunciado

9. Un instituto necesita registrar los cursos y alumnos que estudian en él y necesita un DTD para comprobar los documentos XML de los programas que utiliza:

   - Tiene que haber un elemento raíz <listacursos>. Existen uno o más cursos.
   - Un curso tiene uno o más alumnos
   - Todo alumno tiene un DNI, un nombre y un apellido. Además, puede que tenga segundo apellido o no.
   - Un alumno escoge una o más asignaturas. Toda asignatura tiene un nombre, un atributo código y un profesor.
   - Un profesor tiene un NRP (Número de Registro Personal), un nombre y un apellido (también puede tener o no un segundo apellido).

Escribe un ejemplo de XML válido y la DTD que refleje estos requisitos

10. Crea el DTD que valide un documento XML que almacene recetas de cocina. La estructura es la siguiente:
    a. El elemento principal es la receta
    b. Una receta está formada por un nombre(un texto libre), un listado con uno o más ingredientes y un texto con las instrucciones de elaboración (un texto libre)
    c. Un ingrediente contiene texto libre
    d. Las recetas tienen un atributo que indica el número de raciones
    e. Los ingredientes tienen un atributo opcional con la cantidad del ingrediente
    f. Los ingredientes tienen un atributo opcional con la unidad en la que están expresadas las cantidades.

11. La empresa "Amuebla SA" tiene sucursales en toda España. Su sede central, al igual que el almacén de donde se surten todas las sucursales se encuentra en Madrid.

Cada sucursal consta de una zona de exposición y otra de gestión que no siempre tienen la misma ubicación.

Los pedidos se hacen al almacén de la central y las sucursales reciben los artículos en el departamento de exposición, y el albarán y el pago se remiten al departamento de gestión.

Es necesario conocer la sucursal que realiza cada pedido y sus direcciones para los envíos correspondientes (en el caso de que ambas coincidan solo aparece una). El código de la sucursal está formado por una cadena de 8 caracteres de los cuales el primero es una letra.

Además se reflejarán los siguientes datos de cada pedido:

a. Código del pedido
b. Nombre del trabajador que realiza el pedido.
c. Fecha del pedido.
d. Observaciones sobre el pedido. Si este dato existe sus valores serán: urgente o incompleto.
e. Plazo de revisión de los productos recibidos. Será un intervalo de tiempo expresado en días y dependerá del precio final del pedido.

Para cada uno de los artículos que se incluyen en un pedido se guardará:

Código del artículo, formado por tres letras mayúsculas y tres dígitos separados por un guion. Es la referencia que tiene que dar el ordenante a la sede en caso de devolución de algún artículo.

a. Número de unidades pedidas.
b. Precio de cada unidad.
c. Observaciones del artículo.

Se pide construir:

a. El vocabulario para el documento XML que utiliza esta empresa para gestionar los pedidos utilizando un DTD externo.
b. Realizar un fichero XML que se corresponda con una instancia del vocabulario diseñado y asociarle al DTD.
c. Modificar ese fichero XML para asociarle al esquema diseñado.

12. Dado el siguiente código XML, cree un DTD que siga esa estructura teniendo en cuenta que la agenda puede tener 1 o más contactos y que todos los elementos son obligatorios excepto email y casado. Tanto teléfono como casado son elementos vacíos.

```
<?xml version="1.0" encoding="utf-8" ?>
<agenda>
  <contacto>
      <nombre> AlumnoLM </nombre>
      <ciudad cp="29000">Málaga</ciudad>
      <telefono numero="555985421"/>
      <email>prueba@pruebas.com</email>
      <casado v="no"/>
  </contacto>
</agenda>
```

13. El siguiente documento XML está bien formado. Sin embargo, no es válido. Para que lo sea, realiza los cambios necesarios en la DTD interna de dicho documento.

```
<?xml version-"1.0" encoding-"UTF-8"?>
<!DOCTYPE declaraciones [
   <!ELEMENT declaraciones (consonante | tipo_de_dato |
variable)>
   <!ELEMENT consonante (#PCDATA)>
      <!ATTLIST consonante valor CDATA #REQUIRED>
   <!ELEMENT tipo_de_dato (#PCDATA)>
   <!ELEMENT variable (#PCDATA)>
      <!ATTLIST variable tipo CDATA #REQUIRED>
]>

<declaraciones>
   <consonante valor="3.141592">PI</consonante>
   <variable tipo="real">radio</variable>
   <variable>area</variable>
   <variable>longitud</variable>
</declaraciones>
```

14. El siguiente documento XML está bien formado. Sin embargo, no es válido. Para que lo sea, realiza los cambios necesarios en dicho documento, pero sin modificar la DTD interna.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE formas [
   <!ELEMENT formas (cuadrado, triangulo, circulo, otra)>
   <!ELEMENT cuadrado (#PCDATA)>
      <!ATTLIST cuadrado lados CDATA #REQUIRED>
   <!ELEMENT triangulo (#PCDATA)>
      <!ATTLIST triangulo lados CDATA #REQUIRED>
   <!ELEMENT circulo (#PCDATA)>
   <!ELEMENT otra EMPTY>
      <!ATTLIST otra lados CDATA #REQUIRED>
]>

<formas>
   <cuadrado lados="4">tablero</cuadrado>
   <circulo>anilla</circulo>
   <triangulo>señal</triangulo>
   <otra lados="7"/>
</formas>
```

15. Dado el siguiente DTD crear un fichero XML que sea válido para él, y compruébelo con un procesador (por ejemplo XML Copy Editor). Copie el código en un fichero llamado “newspaper.dtd” y haga referencia a él en el XML.

```
<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>
```

16. Construye un DTD que se ajuste a la siguiente jerarquía de datos ,XML en forma de árbol: