Bases de Datos
Práctica de XML

Ejercicio 1: Dar una DTD para definir el vocabulario correspondiente a las siguientes informaciones sobre alumnos:

```
Alumno = (nombre, conjunto-hijos setof(hijos), conjunto-materias
setof(materias))
Hijos = (nombre, cumpleaños)
Cumpleaños = (dia,mes,año)
Materias = (nombre, tipo,ConjuntoExámenes setof(Examenes))
Exámenes = (año, nota, Universidad)
Amigos = (Universidad, conjunto-amigos setof(Alumno))
Tipo = terciario | grado | postgrado
```

Ejercicio 2: Escribir las siguientes consultas en XQuery asumiendo el DTD del ejercicio 1.

1. Encontrar los nombres de todos los alumnos que tienen un hijo cuyo cumpleaños cae en marzo.
2. Encontrar aquellos alumnos que se examinaron en la materia "Bases de Datos.en la Universidad Nacional de Córdoba.
3. Listar todos los nombres de materias que corresponden a algún alumno.
4. Generar un reporte donde para cada alumno se muestra las lista de los nombres de sus amigos. Para cada amigo se muestra la información de si tiene hijos, la cantidad de materias en que está involucrado y del promedio general de todos sus exámenes.

Ejercicio 3: Considerar la representación XML de información bancaria de abajo. Se pide:

1. Mostrar la representación árbol de los datos XML.
2. Dar una representación alternativa para la misma información que contenga los mismos datos, pero usando atributos en lugar de subelementos.
3. Dar el DTD para esta representación.
4. Escribir una consulta XQuery para encontrar el saldo total en cada sucursal a partir de todas las cuentas (ayuda: se puede usar una consulta anidada para lograr el efecto de **group by** de SQL).
5. Escribir una consulta en XQuery para calcular la reunión externa por la izquierda de los elementos *impositor* con los elementos *cuenta* (ayuda: se puede usar la cuantificación universal).

```
<banco>
        <cuenta>
                <número-cuenta> C101 </número-cuenta>
                <nombre-sucursal> Centro </nombre-sucursal>
                <saldo> 500 </saldo>
        </cuenta>
        <cuenta>
                <número-cuenta> C102 </número-cuenta>
                <nombre-sucursal> Navacerrada </nombre-sucursal>
                <saldo> 400 </saldo>
        </cuenta>
        <cuenta>
                <número-cuenta> C201 </número-cuenta>
                <nombre-sucursal> Galapagar </nombre-sucursal>
```

```
        <saldo> 900 </saldo>
    </cuenta>
    <cliente>
        <nombre-cliente> González </nombre-cliente>
        <calle-cliente> Arenal </calle-cliente>
        <ciudad-cliente> La Granja </ciudad-cliente>
    </cliente>
    <cliente>
        <nombre-cliente> López </nombre-cliente>
        <calle-cliente> Mayor </calle-cliente>
        <ciudad-cliente> Córdoba </ciudad-cliente>
    </cliente>
    <impositor>
        <número-cuenta> C101 </número-cuenta>
        <nombre-cliente> González </nombre-cliente>
    </impositor>
    <impositor>
        <número-cuenta> C201 </número-cuenta>
        <nombre-cliente> González </nombre-cliente>
    </impositor>
    <impositor>
        <número-cuenta> C102 </número-cuenta>
        <nombre-cliente> López </nombre-cliente>
    </impositor>
</banco>
```

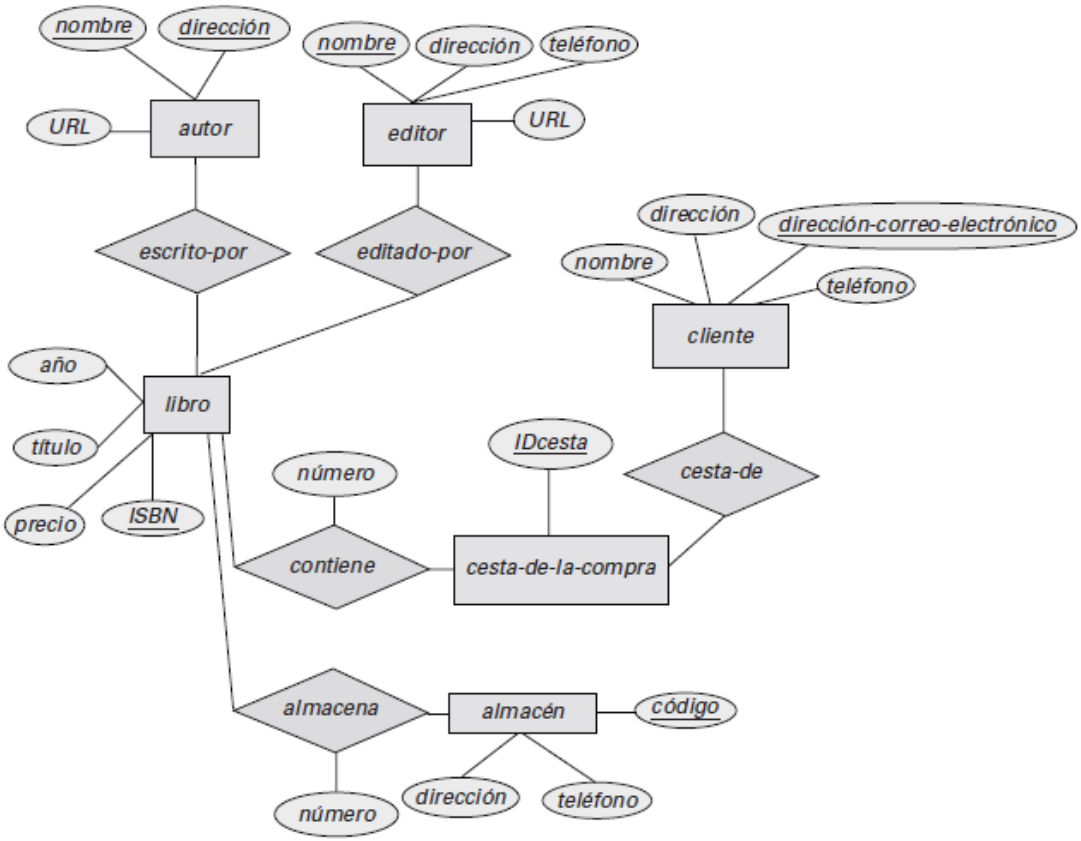
Ejercicio 4: Se tiene el siguiente DTD sobre bibliografía:

```
<!DOCTYPE bibliografía [
<!ELEMENT libro (título, autor+, año, editor, lugar?, ISBN)>
<!ELEMENT artículo (título, autor+, revista, año, número, volumen,
páginas?)>
<!ELEMENT autor (apellido+, nombre)>
<!ELEMENT título (#PCDATA)>
<!ELEMENT año (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT revista (#PCDATA)>
<!ELEMENT número (#PCDATA)>
<!ELEMENT volumen (#PCDATA)>
<!ELEMENT páginas (#PCDATA)>
<!ELEMENT apellido (#PCDATA)>
<!ELEMENT nombre (#PCDATA)>
]>
```

Escribir las siguientes consultas XQuery:

1. Encontrar todos los autores que tienen un libro y un artículo en el mismo año.
2. Mostrar los libros y artículos ordenados por año.
3. Mostrar libros con más de un autor.

Ejercicio 5: Dése la DTD para una representación XML de la información de la siguiente figura. Crear un tipo de elemento separado para representar cada relación, pero úsense ID e IDREF para implementar las claves primarias y externas.



Ejercicio 6: se tiene un DTD correspondiente a accidentes de autos. La descripción de la estructura del archivo XML es como sigue: tenemos personas con su DNI, nombre (primer nombre y apellido) y dirección (calle, número, número de apartamento o nada si no es departamento, ciudad, código postal), tenemos los autos, cada uno con su matrícula, número, modelo y número de DNI del dueño. Tenemos accidentes donde cada accidente consiste de horario del mismo (fecha – dia, mes y año - y hora en que ocurrió), los autos involucrados en el accidente, indicando para cada uno de ellos la matrícula y quién era el conductor del coche, y finalmente se tiene el monto del daño en el accidente. Realizar el DTD; para el mismo se pide usar todo lo posible atributos de tipo ID, IDREF e IDREFS.

Realizar las siguientes consultas en XQuery:

a) Mostrar el DNI y nombre de todas las personas que son dueñas de más de 2 coches ordenando los resultados por nombre en forma decreciente (primero por el apellido y luego por el primer nombre).

b) Listar para cada fecha del año (en que ocurrieron accidentes) la cantidad de accidentes y el monto acumulado en daños en la misma. Considerar los años ordenados forma creciente y las fechas en forma decreciente.

Ejercicio 7: Resolver.

a) Explicar porqué XQuery es más expresivo que XPath. Mencionar al menos cuatro facilidades para hacer consultas que permite XQuery y no XPath.

b) Se desea mantener para una empresa aérea, una pequeña base de datos en archivos XML. La información necesaria es: vuelo número, horario de partida, horario de partida efectivo, horario de llegada, horario de llegada efectivo, número de pasajeros, precio y servicios a bordo: primera clase, económica. Se pide:

1) Escribir un DTD para representar el esquema de la información
2) En base al DTD, escribir un documento XML de ejemplo que lo respete.
3) Expresar las siguientes consultas usando XQuery
   1. Calcular el monto facturado por la empresa en el mes de octubre.
   2. Calcular el número de vuelos que salieron con demora.
   3. Calcular la cantidad de personas que han estado demoradas por más de dos horas.

Nota: puede usar la función built-in `month-from-dateTime`; además para no tener que trabajar con operaciones entre horarios y duraciones, puede suponer que hay una función `hoursBetween($t1 as xs:dateTime,$t2 as xs:dateTime) as xs: integer`.

Ejercicio 8: Se quiere utilizar una base de datos XML para almacenar artículos periodísticos que se venderán a clientes a través de un portal Web. Se hará una lista de documentos que se quieren comprar utilizando búsquedas sobre los datos en los documentos. Suponga que los archivos XML son como el siguiente:

```
<biblioteca>
      <autor id=”au001” nombre= “Pepe Rodríquez” ...>
            <articulo id= “ar001” .... >
                  <titulo> Comparativa de BBDD XML nativas y ...</titulo>
            </articulo>
            <articulo id= “ar002” .... >
                  <titulo> Propagación de restricciones ....</titulo>
            </articulo>
      </autor>
      <autor id=”au002” ...>
            <articulo id=“ar003” .... >
...
</biblioteca>
```

1. Escribir un documento DTD que valide documento XML de arriba.
2. Dar la expresión XPath que corresponde a los artículos escritos por “Pepe Rodríguez”.
3. Resolver usando XQuery: producir un reporte que muestra primero los datos personales de “Pepe Rodríguez”; luego muestra la cantidad de artículos escritos por él; y finalmente muestra todos los títulos de los artículos escritos por esa persona, todos ordenados de mayor a menor.