

## Trabajo de primer bimestre

**Fecha límite: Viernes, 29 de mayo de 2015 23:59**

Este trabajo tiene como objetivo perfeccionar el manejo de documentos XML desde distintas situaciones que pueden presentarse. El material visto en clase será suficiente para solucionar los ejercicios 1 y 2. Para el ejercicio 3 se necesitará investigar sobre el elemento import de un XML Schema.

Todos los documentos XML deberán ser validados contra su respectivo XML Schema. Para esto se utilizará la clase Java incluida en el material "DomValidator.java" que se utilizará de la siguiente manera:

```
java DomValidator archivo.xml schema.xsd [schema2.xsd schema3.xsd]
```

Ejemplo:

```
java DomValidator ejercicio2_personas.xml ejercicio2_personas.xsd
```

```
java DomValidator ejercicio3_alumno.xml ejercicio3_persona.xsd  
ejercicio3_docencia.xsd ejercicio3_alumno.xsd
```

### Ejercicio 1: Diseñar un XML a partir de una tabla de datos.

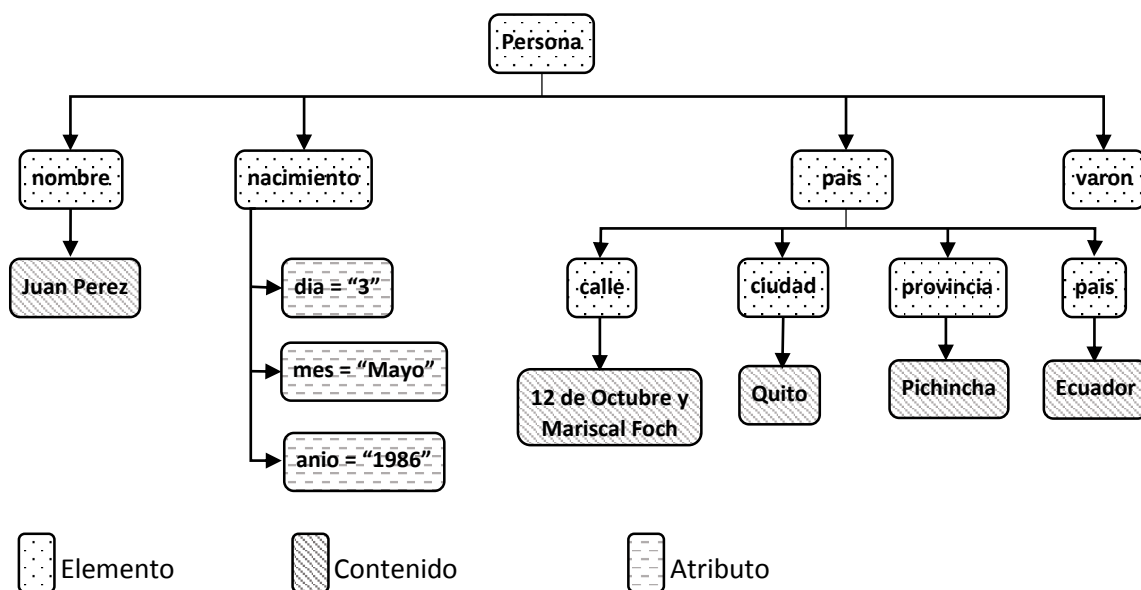
Representar la siguiente tabla en un documento XML bien formado que contenga la siguiente información:

LIBROS					
Título	Autor	Área	Editorial	Precio	Año de publicación
Cien años de soledad	Gabriel García Márquez	Nóvela	Editorial 1	\$10.00	1960
El señor de los anillos	J. R. R. Tolkien	Fantasía	Editorial 2	\$15.46	2001
SOA Simple	Lonneke Dikmans	Técnica	Editorial 3	\$45.78	2015
Nacho Lee	Nacho	Lectura	Editorial 2	\$11.23	1986
22/11/1963	Stephen King	Novela	Editorial 4	\$5.63	2009
Hamlet	William Shakespeare	Novela	Editorial 1	\$23.51	1920

## Ejercicio 2: Obtener XML Schema a partir de una jerarquía de datos.

a) Proporcionar un XML Schema que refleje la jerarquía mostrada en la figura suponiendo que “nacimiento” es un elemento opcional y hay al menos una dirección y teniendo en cuenta los siguientes requisitos:

- En un documento solo puede aparecer el elemento hombre o mujer pero no ambos.
- Los valores del atributo “dia” están comprendidos entre 1 y 31. Definir el tipo “tipoDia”.
- Los valores del atributo “mes” son de tipo cadena y tienen que coincidir con uno de los meses del año. Definir el tipo “tipoMes”.
- Los valores del atributo “anio” están comprendidos entre 1900 y 2015. Definir el tipo “tipoAnio”.
- Los valores de “nombre”, “calle”, “ciudad”, “provincia” y “pais” tienen como máximo 50 caracteres. Definir el tipo “tipoNombre”.



b) Escribir un documento XML instancia que referencie al Schema y contenga los siguientes datos, validar el documento XML con DomValidator.java:

- Juan Perez.  
Fecha de nacimiento: 3 de mayo de 1986  
Dirección: 12 de Octubre y General Foch, Quito, Pichincha, Ecuador
- María López.  
Dirección1: Tamayo n12 y Francisco Salazar, Quito, Pichincha, Ecuador  
Dirección2: Pichincha 605 entre Clemente Ballén y 10 de Agosto, Guayaquil, Guayas, Ecuador

### Ejercicio 3: Definir un XML Schema y un documento XML a partir de requisitos.

El documento XML (“alumno.xml”) muestra la estructura de documentos que contienen información de alumnos de un instituto. De cada alumno se guardan datos personales y datos académicos. Para ello se van a definir tres XML Schemas: personas, docencia y alumno (que deben estar basados en la estructura de alumno.xml) y que se explican a continuación.

**personas.xsd:** contiene las definiciones de elementos correspondientes a datos personales.

**docencia.xsd:** contiene las definiciones de elementos relacionados con el pensum y las asignaturas.

En este esquema se definen los siguientes elementos:

- pensum: tipo cadena para guardar el nombre de un pensum
  - asignatura: cada asignatura tiene los siguientes elementos hijo:
    - nombre de la asignatura
    - horas semanales
    - profesor: el nombre del profesor que imparte la asignatura
- 
- a) Escribir los dos esquemas “personas.xsd”, “docencia.xsd” y “alumno.xsd” basándose en uno de los patrones de diseño vistos en clase (indicar el patrón elegido) y en la estructura del documento alumno.xml
  - b) Asociar los espacios de nombres destino “http://www.trabajo.com/persona”, “http://www.trabajo.com/docencia” y “http://www.trabajo.com/alumno” respectivamente.
  - c) En todos los casos se debe especificar que en los documentos instancia los elementos y atributos pertenecientes al espacio de nombres vayan cualificados con un prefijo.
  - d) El XML Schema “alumno.xsd” debe utilizar los dos Schemas definidos, y debe validar el documento “alumno.xml”. Para este punto será necesario investigar y utilizar el elemento import dentro de “alumno.xsd”. Una buena referencia sobre este elemento se la puede encontrar en: <http://www.xfront.com/ZeroOneOrManyNamespaces.html>

### alumno.xml

```
<?xml version="1.0" encoding="utf-8"?>
<alu:alumno xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="
    http://www.trabajo.com/persona persona.xsd
    http://www.trabajo.com/docencia docencia.xsd
    http://www.trabajo.com/alumno alumno.xsd"
    xmlns:per="http://www.trabajo.com/persona"
    xmlns:doc="http://www.trabajo.com/docencia"
    xmlns:alu="http://www.trabajo.com/alumno">

    <per:persona per:nacimiento="1999-10-20">
        <per:datos>
            <per:nombre>Pepe</per:nombre>
            <per:apellidos>Garcia</per:apellidos>
            <per:dni>25390952X</per:dni>
        </per:datos>
        <per:comentario>repetidor</per:comentario>
    </per:persona>

    <datos_academicos>
        <doc:pensum>1ASIR</doc:pensum>

        <doc:asignatura>
            <doc:nombre>Lenguajes de Marcas</doc:nombre>
            <doc:horas>4</doc:horas>
            <doc:profesor>Juan Moreno</doc:profesor>
        </doc:asignatura>

        <doc:asignatura>
            <doc:nombre>Fundamentos del Hardware</doc:nombre>
            <doc:horas>3</doc:horas>
            <doc:profesor>Virginia López</doc:profesor>
        </doc:asignatura>
    </datos_academicos>
</alu:alumno>
```