

## *XML*

### *TP N° 2: Validación con esquemas - DTD*

Para parsear un documento xml con una estructura DTD asociada recordar utilizar el parámetro **-V**

```
java dom.Writer -v nombreDocumento.xml
```

#### *Ejercicio 1*

Definir **emails.dtd** que permita validar el documento emails.xml creado en el ejercicio 2 del TP 1. Indicar y justificar las validaciones que no se pueden realizar.

#### *Ejercicio 2*

Cambiar al documento **emails.xml** (incluyendo su DTD asociado) para que pueda también representar las respuestas de un correo electrónico. Sugerencia: Utilizar ID e IDREF para indicar un correo electrónico y sus respuestas asociadas, solamente en los casos necesarios. Es decir, si un correo electrónico no tiene respuestas asociadas, no debería tener un ID.

#### *Ejercicio 3*

- 3.1. Cambiar al documento **emails.xml** para que además un mismo correo electrónico pueda ser enviado a más de una persona.
- 3.2. Agregar al elemento **header** del documento un atributo que indique el encoding (UTF-8, UTF-16, ISO-8859-1), cuyo valor por defecto sea **UTF-8**. No confundir el encoding del documento xml con el encoding del correo electrónico. Además, agregar otro atributo indicando el cliente de correo electrónico utilizado por el remitente. Agregar las validaciones necesarias al archivo **emails.dtd**. ¿Importa el orden en que se agregan los atributos al tag? ¿Importa el orden en que se agregan las validaciones de los atributos?
- 3.3. Agregar entre los elementos **header** y **subject** un código de seguridad alfa numérico de 10 caracteres. Actualizar el archivo **emails.dtd** para reflejar que el elemento **email** tiene contenido mixto (sus sub-elementos y el texto). Validar el orden y la cardinalidad de los elementos. Explicar el resultado.
- 3.4. Realizar una representación visual (en papel) de la estructura final del documento, sin su contenido.

## Ejercicio 4

Indicar si las siguientes afirmaciones son Verdaderas o Falsas

- 4.1. Un DTD puede declarar más de un atributo en distintos elementos que sean ID obligatorios. Por ejemplo, el atributo “identificador” que sea ID y que esté en más de un elemento (diferentes tipos de elementos).
- 4.2. Un DTD puede declarar más de un atributo en distintos elementos que sean ID obligatorios, y cada atributo tiene un nombre distinto. Por ejemplo, el atributo “identificador1” en el elemento “elemento1” y el atributo “identificador2” en el elemento “elemento2”.
- 4.3. Sea un DTD con 2 atributos A y B, ambos de tipo ID y #required. Cada uno de dichos atributos se utilizan en distintos elementos. Es posible que exista un XML que valide con dicho DTD tal que el conjunto de todos los valores que toma en el documento XML el atributo A y el conjunto de todos los valores que toma en el documento XML el atributo B sean no disjuntos. Si es V, exhibir un XML que valide con dicho DTD y discutir el riesgo de esto.

## Ejercicio 5

RSS es una familia de formatos de documentos XML, utilizado cada vez más activamente para el intercambio de información y noticias entre sitios Web (B2B) y entre sitios Web y usuarios.

Es este último caso, la intención es permitir que el usuario, a través de un lector RSS, pueda recibir información actualizada de sus páginas favoritas, sin tener que visitarlas una a una.

Por ejemplo, algunos readers online pueden ser:

DiggReader: <https://digg.com/reader/>

Feedly: <http://feedly.com/>

El término RSS técnicamente describe a 3 estándares:

- Rich Site Summary (RSS 0.9.1)
- RDF Site Summary (RSS 0.9 y 1.0)
- Really Simple Syndication (RSS 2.0), siendo este último el más utilizado hoy en día para la provisión de información actualizada desde un sitio web a sus “suscriptores”

Se denomina a los flujos de datos que se proveen bajo este mecanismo: fuentes RSS (o *RSS feeds*).

RSS 2.0 es un estándar que especifica cómo deben formarse los documentos XML a proveer. A continuación trabajaremos con un subconjunto del estándar, tomando las siguientes reglas:

1. El documento debe tener como nodo raíz **rss** y un atributo obligatorio **version** que indique la versión de RSS utilizada, siendo los valores posibles: 0.9.1, 0.9, 1.0, 2.0.
2. Siguiendo al elemento raíz, se tiene un único elemento **channel** con información sobre el canal emisor del documento XML y su contenido
3. Los elementos requeridos del **channel** son:
  1. **title** nombre del canal
  2. **link** una URL válida al sitio Web relacionado con el canal
  3. **description** descripción breve del canal
4. Los elementos opcionales del **channel** son:
  1. **pubDate** fecha de publicación del contenido (según el RFC #822, <http://asg.web.cmu.edu/rfc/rfc822.html> )
  2. **webMaster** dirección de correo electrónico válida del responsable del sitio Web.



3. **language** indica el idioma y dialecto en que está generado el contenido. Consideraremos sólo dos posibilidades: es-ar ó en-us
4. Una cantidad indeterminada de elementos **item** donde cada elemento representa un artículo o noticia. Cada elemento **item** posee:
  1. **title** título del artículo o noticia
  2. **link** URL válida que lleva a la noticia en Internet
  3. **pubDate** idem pubDate en channel

5.1. Construir un documento **noticias.xml** con al menos tres noticias/artículos y que esté bien formado.

5.2. Definir **noticias.dtd** que permita validar el documento anterior. Indicar y justificar las validaciones que no se pueden realizar.

5.3. Buscar al menos 3 ejemplos en Internet de sitios de noticias o blogs e inspeccionar si proveen RSS Feeds y su formato. ¿Cómo nos indican los browsers que un sitio provee un Feed RSS?