

## Ejercicio 1

Dado este documento descrito en SGML, usando el tipo de documento urjc\_ml definido en el fichero urjc\_ml.dtd

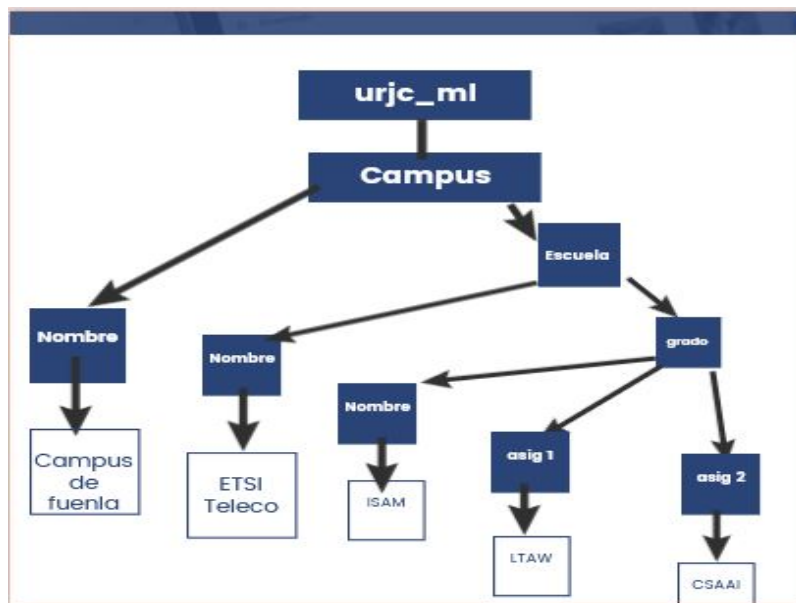
```
<!DOCTYPE urjc_ML SYSTEM "urjc_ml.dtd">
<urjc_ml>
  <!-- include URL URJC -->
  <campus>
    <nombre>Campus de Fuenlabrada</nombre>
    <escuela>
      <nombre>ETSI Telecomunicación</nombre>
      <grado>
        <nombre>Ingeniería en Sistemas Audiovisuales y Multimedia </nombre>
        <asignatura>Laboratorio de Tecnologías Audiovisuales en la Web</asignatura>
        <asignatura>Construcción de servicios y Aplicaciones Audiovisuales en Internet</asignatura>
      </grado>
    </escuela>
  </campus>
</urjc_ml>
```

- a) Explica para qué sirve la primera línea del documento y por qué es necesaria
  - La primera línea del documento es un lenguaje propio de marcado. Esta línea sirve para:
    - permitirnos crear nuestro propio lenguaje de marcado adaptado a nuestras necesidades.
    - Así, especifica la sintaxis de las etiquetas y nos permite definir sus nombres y su estructura como nosotros queramos. El fichero donde definimos qué etiquetas son válidas y su sintaxis, se denomina definición del tipo de documento y su extensión es .dtd (DTD = Document Type Definition)
  - Además, Usando SGML escribimos la definición de nuestro lenguaje en el fichero "urjc\_ml.dtd", es decir, tenemos que saber que la definición de las etiquetas del lenguaje OBML está en ese fichero
- b) Si se omitiese esta línea, ¿Qué piensas que ocurriría?
  - No podríamos crear nuestro propio lenguaje de marcado
- c) Sin conocer el contenido del fichero urjc\_ml.dtd, ¿el documento es sintácticamente correcto?
  - Si, ya que termina en dtd.
- d) ¿Qué hace la tercera línea?
  - Es un comentario
- e) ¿Cuántas etiquetas de apertura hay? ¿Cuántas de cierre?

Etiquetas	numero
apertura	10
cierre	9

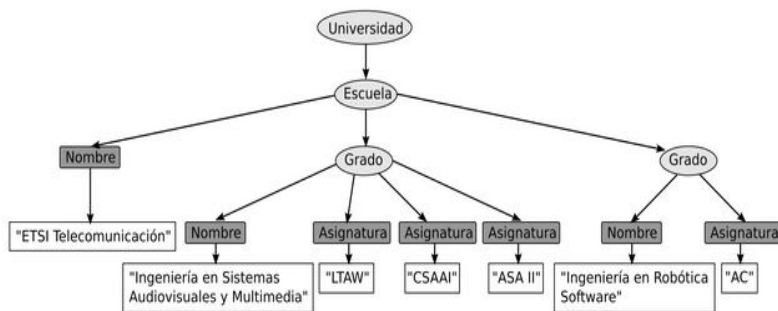
- f) Dibuja el diagrama de contenedores de este documento

- Llamamos diagrama de contenedores o diagrama de cajas a la representación de la estructura, así unos elementos contienen a otros elementos.
- g) Dibuja la estructura en árbol que define este documento



## Ejercicio 2

Este árbol representa la estructura del tipo de documento universidad, que está definido en el archivo llamado universidad.dtd. Los nombres de las etiquetas de cada elemento del árbol están escritos en los nodos. Estos nombres son válidos y se encuentra definidos dentro del documento DTD



- a) Escribe el documento en SGML que representa esa estructura

```

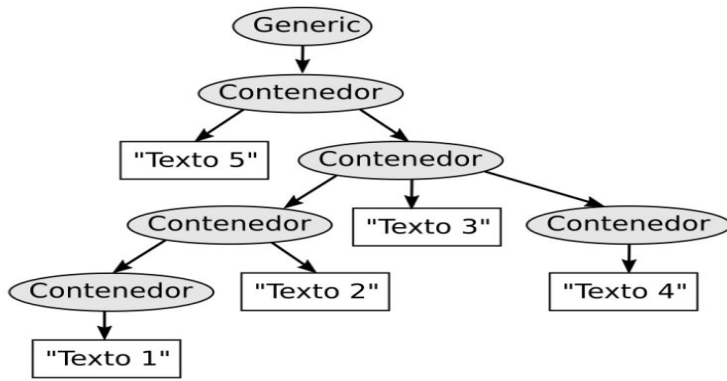
<!DOCTYPE urjc_ML SYSTEM "universidad.dtd">
<urjc_ML>
  <universidad>
    <escuela>
      <nombre>ETSI Telecomunicación</nombre>
      <grado>
        <nombre>Ingeniería en Sistemas Audiovisuales y Multimedia</nombre>
        <asignatura>LTAW</asignatura>
        <asignatura>CSAII</asignatura>
        <asignatura>ASA II</asignatura>
      </grado>
      <grado>
        <nombre>Ingeniería en Robótica y Software</nombre>
        <asignatura>AC</asignatura>
      </grado>
    </escuela>
  </universidad>
</urjc_ML>

```

- b) ¿Cuántos elementos contenedores hay? Indica sus nombres
  - Hay 18:
    - Universidad
    - Escuela
    - 2 de grado
    - 7 asignatura/nombre
    - 7 de texto
- c) ¿Cuántos elementos no terminales hay? Indica sus valores
  - Elementos no terminales hay 7 y son:
    - los 7 de texto
  - Elementos terminales serían 11 y son:
    - Universidad
    - Escuela
    - 2 de grado
    - 7 asignatura/nombre
- d) ¿Cuántos elementos hay en el nivel 3?. Escribe sus nombres
  - El nivel 3 serían los siguientes:
    - Nombre
    - asignatura
    - asignatura
    - asignatura
    - Nombre
    - asignatura

## Ejercicio 3

Este árbol representa la estructura del tipo de documento genérico, que está definido en el archivo llamado generic.dtd. Los nombres de las etiquetas de cada elemento del árbol están escritos en los nodos. Estos nombres con válidos y se encuentra definidos dentro del documento DTD



- a) Escribe el documento en SGML que representa esa estructura

```

1  <!DOCTYPE urjc_ML SYSTEM "Generic.dtd">
2  <urjc_ML>
3  <Generic>
4    <contenedor>
5      "Texto 5"
6    </contenedor>
7    <contenedor>
8      "Texto 3"
9      <contenedor> "Texto 4" </contenedor>
10     <contenedor> "Texto 2"
11       <contenedor> "Texto 1" </contenedor>
12     </contenedor>
13   </contenedor>
14 </Generic>
15 </urjc_ML>

```

- b) ¿Cuántos elementos hay en total?
  - 11 elementos
- c) ¿Cuántos elementos terminales hay?. Indica sus valores
  - Hay 6 elementos terminales y son:
    - Generic
    - Contenedor
    - Contenedor
    - Contenedor
    - Contenedor
    - Contenedor
- d) ¿Cuántos elementos no terminales hay?. Indica cuántos hay en cada nivel
  - Hay 5 elementos no terminales y son:
    - Texto 5
    - Texto 3
    - Texto 4
    - Texto 2
    - Texto 1
- e) ¿Cuántos elementos hay en el nivel 5?. Indica sus nombres
  - Solo uno:
    - Texto 1

## Ejercicio 4

Dado el siguiente documento escrito en markdown, dibuja:

```
# FPGAs

## FPGAs libres

### Familias

* ice40
* UP5K
* ECP5

### Placas

1. Alhambra II
2. Icestick
3. ULX3S
4. iceBreaker
5. TinyFPGA
```

- a) Su diagrama de contenedores
- b) Su estructura en árbol