

EJERCICIO 1.- Indica por qué los siguientes ejemplos XML son incorrectos.

Ejercicio 1a)

```
<elemento>Elemento 1</elemento>
<elemento>Elemento 2</elemento>
```

Sería incorrecta debido a que no puede haber 2 etiquetas de raíz o falta una etiqueta raíz principal que albergue estas dos.

Dado que lo que se intuye que con esta instrucción lo que se pretende es la creación de 1 elemento raíz que a su vez contenga dos elementos hijos, la forma correcta de argumentar esta instrucción sería: `<!ELEMENT elemento (elemento1, elemento2)>`

Dónde `<!ELEMENT` es la declaración del elemento, `elemento` es el nombre del elemento raíz y `(elemento1, elemento2)` son los elementos que derivan del elemento raíz.

Ejercicio 1b)

```
<elemento>Coche</ELEMENTO>
```

Sería incorrecto porque los nombres de las etiquetas al inicio y final han de ser idénticas y en este caso la etiqueta inicio está en minúscula y la etiqueta final en mayúsculas. La forma correcta de argumentar esta instrucción sería: `<elemento>Coche</elemento>` o `<ELEMENTO>Coche</ELEMENTO>`.

Ejercicio 1c)

```
<negrita><italica>Texto XML</negrita></italica>
```

Es incorrecto dado que la etiqueta `italica` debe de cerrarse en primer lugar. La forma correcta de argumentar esta instrucción sería: `<negrita><italica>Texto XML</italica></negrita>`.

Ejercicio 1d)

```
<rango>1 < 5 & 11 > 7</rango>
```

Es incorrecta porque dentro de la etiqueta no se puede contener los símbolos `<`, `>`, `&`. Para ello habría que utilizar las siguientes cadenas `>` para `>` `<` para `<` y `&` para `&`

De forma que la forma correcta sería la siguiente instrucción:

```
<rango>1 &lt; 5 &amp; 11 &gt; 7</rango>
```

Ejercicio 1e)

```
<XMLfichero>Marcas.xml</XMLfichero>
```

Es incorrecto, ya que el inicio y cierre de los nombres de las etiquetas no pueden contener la cadena "xml" en ninguna versión (mayúsculas o minúsculas).

Ejercicio 1f)

```
<![CDATA[ <[[aa]]>]]>
```

Es incorrecto porque no puede contener la cadena `"]]>` por compatibilidad con SGML

Ejercicio 1g)

```
<user@uo>Pedro@Empleados</user@guo>
```

Es incorrecto ya que el nombre de la etiqueta debe de ser el mismo al inicio y al cierre. Además, no se puede utilizar el carácter especial “@”. Una forma correcta sería por ejemplo la siguiente instrucción: `<user@guo>PedroEmpleados</user@guo>`

Ejercicio 1h)

```
<texto>El titular de hoy se basa en esta <subrayado>noticia</subrayado></texto>
```

Es incorrecto ya que la etiqueta “subrayado” no se cierra correctamente, la instrucción correcta sería: `<texto>El titular de hoy se basa en esta
<subrayado>noticia</subrayado></texto>`

EJERCICIO 2.- Diseñar un documento válido en XML que permita estructurar la información de una agenda de teléfonos, suponer que la información que podemos tener de una persona es su nombre y apellidos, su dirección y sus teléfonos, que pueden ser el teléfono de casa, el móvil y el teléfono del trabajo.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE agenda>
<agenda>
  <identificacion>
    <nombre>David</nombre>
    <apellido>García</apellido>
  </identificacion>
  <direccion>
    <calle>Avenida Filiberto</calle>
    <numero>57</numero>
    <piso>2</piso>
    <letra>A</letra>
    <localidad>Salamanca</localidad>
    <codigopostal>37007</codigopostal>
  </direccion>
  <telefonos>
    <telcasa>923000000</telcasa>
    <telmovil>622000000</telmovil>
    <teltrabajo>923000001</teltrabajo>
  </telefonos>
</agenda>
```

En la primera línea indicamos que es un archivo xml e indicamos su versión, en este caso la versión “1.0”, seguidamente indicamos el tipo de caracteres que puede soportar; en este caso escogí el iso-8859-1 para que se pueda incluir caracteres del alfabeto castellano como la “ñ”. En primer lugar, utilizamos el nodo raíz agenda, en el que van a ir incluidos los nodos secundarios de identificación, dirección y teléfonos, que a su misma vez están compuestos de nodos hijos. De forma que dentro del nodo de identificación estarán los nodos hijos de nombre y apellido. En el caso de dirección se han incluido los nodos hijos de calle, número,

piso, letra, localidad y código postal. Así mismo, dentro del nodo teléfonos integramos otros 3 sub-nodos en los que incluimos el teléfono de casa, el teléfono móvil y el teléfono del trabajo.

EJERCICIO 3.- Diseñar un documento válido en XML que permita estructurar la información de las recetas de cocina de un restaurante y aplicarlo a la siguiente receta de cocina. Hay que hacerlo de modo que un sistema informático pueda realizar búsquedas por ingredientes, cantidad de comensales o nombre de la receta

Sopa de cebolla (4 personas)

Ingredientes:

- 1 Kg. de cebollas.
- 2 l. de caldo de carne.
- 100 gr. mantequilla.
- 1 cucharada de harina.
- 100 gr. de queso emmental suizo o gruyere rallado.
- Pan tostado en rebanadas.
- Tomillo.
- 1 hoja de laurel.
- Pimienta.

Proceso:

- Pelar y partir las cebollas en rodajas finas.
- Rehogarlas con la mantequilla, sal y pimienta a fuego lento hasta que estén transparentes sin dorarse.
- Añadir la harina sin dejar de remover.
- Ponerlo en una cazuela con el caldo, el tomillo y el laurel.
- Dejar cocer a fuego lento durante unos 15 minutos.
- Poner las rebanadas de pan encima, espolvorear el queso y gratinar al horno.

En la primera línea indicamos que es un archivo XML e indicamos su versión, en este caso la versión “1.0”, seguidamente indicamos el tipo de caracteres que puede soportar; en este caso escogí el “UTF-8”. En primer lugar, utilizamos el nodo raíz receta, donde se integran los nodos plato, número de comensales, ingredientes y procesos. Dentro de ingredientes se han añadido los nodos de los diferentes ingredientes de esta receta, incluyendo el nombre de este, la unidad de medida y la cantidad de unidades. Por otra parte, dentro del nodo proceso incluimos todos los pasos de esta receta.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE receta>
  <receta>
    <plato>Sopa de cebolla</plato>
    <comensales>4</comensales>
    <ingredientes>
      <ingrediente cantidad="1" unidad="Kg." nombre="cebollas" />
      <ingrediente cantidad="2" unidad="litros" nombre="caldo de carne" />
      <ingrediente cantidad="100" unidad="gramos" nombre="mantequilla"/>
      <ingrediente cantidad="1" unidad="cucharada" nombre="harina" />
      <ingrediente cantidad="100" unidad="gramos" nombre="queso
        emmental suizo o gruyere rallado" />
      <ingrediente cantidad="" unidad="" nombre="Pan tostado en
        rebanadas" />
      <ingrediente cantidad="" unidad="" nombre="Tomillo" />
      <ingrediente cantidad="1" unidad="hoja" nombre="Laurel" />
      <ingrediente cantidad="" unidad="" nombre="Pimienta" />

    <proceso>
      <paso1>Pelar y partir las cebollas en rodajas finas. </paso1>
      <paso2>Rehogarlas con la mantequilla, sal y pimienta a fuego lento hasta
        que estén transparentes sin dorarse.</paso2>
      <paso3>Añadir la harina sin dejar de remover.</paso3>
      <paso4>Ponerlo en una cazuela con el caldo, el tomillo y el
        laurel.</paso4>
      <paso5>Dejar cocer a fuego lento durante unos 15 minutos.</paso5>
      <paso6>Poner las rebanadas de pan encima, espolvorear el queso y
        gratinar al horno.</paso6>
    </proceso>
  </receta>

```

EJERCICIO 4.- Diseñar un documento XML válido que permita estructurar la información para permitir su gestión informática de los alumnos de un módulo del ciclo formativo DAM. Aplicarlo al módulo de Lenguajes de Marcas y Sistemas de Gestión de Información sabiendo que tiene asignadas 4 horas semanales y es de carácter obligatorio. El módulo se imparte entre el 15 de septiembre de 2010 y el 30 de junio de 2011. Hay matriculados dos alumnos:

- Ana Fernández Gutiérrez con nif 16965696L teléfono 789654321 email ana.fdezgtrez@hotmail.com, su dirección es C/ El Percebe, 13 de Santander CP 39302 No hay información sobre las faltas de asistencia o sus notas hasta el momento.
- Pepito Grillo con nif 98765432H teléfono 656566555 email yhyh@yahoo.com, su dirección es Avd. El Pez, 5 de Suances CP 39401. Su nota es "apto" y no tiene faltas de asistencia

En la primera línea indicamos que es un archivo XML e indicamos su versión, en este caso la versión "1.0", seguidamente indicamos el tipo de caracteres que puede soportar; en este caso escogí el "UTF-8". En primer lugar, utilizamos el nodo raíz curso especificando su nombre, en este caso "DAM". Dentro de este nodo, se incluyen los nodos hijos de modulo, fecha de inicio, fecha de finalización y alumnos.

Dentro del nodo hijo modulo especificamos su nombre: "Lenguajes de Marcas y Sistemas de Gestión de la Información", su carácter "obligatorio" y el número de horas semanales: "4". Dentro de los nodos hijos de fecha de inicio y fecha de finalización se ha incluido dicha información.

Por otra parte, dentro del nodo hijo "alumnos" se han incluido 2 sub-nodos hijos de ficha de alumno, los que a su vez se integran por los nodos nombre, apellido 1, apellido 2, nif, teléfono, email, dirección (integrado por calle, numero, localidad y cp.), faltas de asistencia y notas.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<curso ciclo_formativo="DAM">
  <modulo nombre="Lenguajes de Marcas y Sistemas de Gestión de la Información"
caracter="Obligatorio" horas_semanales="4"/>
  <fecha_inicio> "15 de septiembre de 2010" </fecha_inicio>
  <fecha_final> "30 de junio de 2011" </fecha_final>
  <alumnos>
    <ficha_alumno nombre="Ana" apellido1="Fernández" apellido2="Gutiérrez"/>
      <nif>16965696L </nif>
      <telefono>789654321</telefono>
      <email>ana.fdezgtrrez@hotmail.com</email>
      <dirección calle="C/ El Percebe" numero="13" localidad="Santander" cp="39302"/>
      <faltas_asistencia> No hay información </faltas_asistencia>
      <notas> No hay información </notas>
    <ficha_alumno nombre="Pepito" apellido1="Grillo" apellido2=""/>
      <nif>98765432H</nif>
      <telefono>656566555</telefono>
      <email>yhyh@yahoo.com</email>
      <dirección calle="Avd. El Pez" numero="5" localidad="Suances" cp="39401"/>
      <faltas_asistencia> Sin faltas </faltas_asistencia>
      <notas> Apto </notas>
  </alumnos>
</curso>
```