UT4. 01_XML. ALMACENAMIENTO DE DATOS

Curso: 2DAW.

Lenguajes de Marcas y sistemas de gestión de información

Índice

1. Introducción

Introducción

- Formato de almacenamiento de información a base de etiquetas o marcas definidas por el usuario.
- Reglas ➔documento XML esté **bien formado.**
- Mecanismos para **validar** un documento XML.
  - Atributos que pueden aparecer (vocabulario)
  - Orden en el que aparecen
  - Elementos contiene a elementos
  - Atributos de un elemento
  - Elementos o atributos optativos u obligatorios, etc.

2. Documentos XML

Documentos XML

- **XML - eXtensible Markup Language -** Lenguaje de Marcado eXtensible.
- Metalenguaje de marcas –sin conjunto fijo de etiquetas-.
- Define una sintaxis general para maquetar datos con etiquetas sencillas y comprensibles.
- Proporciona un formato estándar para documentos informáticos.
- Es un formato flexible

Documentos XML

- No es…
  - Lenguaje de programación
  - Protocolo de comunicación
  - Sistema gestor de bases de datos. Una base de datos relacional puede contener campos del tipo XML. Existen, incluso, bases de datos XML nativas, que todo lo que almacenan son documentos con formato XML.
  - No es propietario (no pertenece ninguna compañía).

Documentos XML

- Formato de texto plano, adecuado para almacenar información y trasmitirla
- Son relativamente ligeros para ser almacenados y enviados.
- Extensión .xml, aunque no imprescindible.
- Etiquetas = metainformación → información sobre la información.

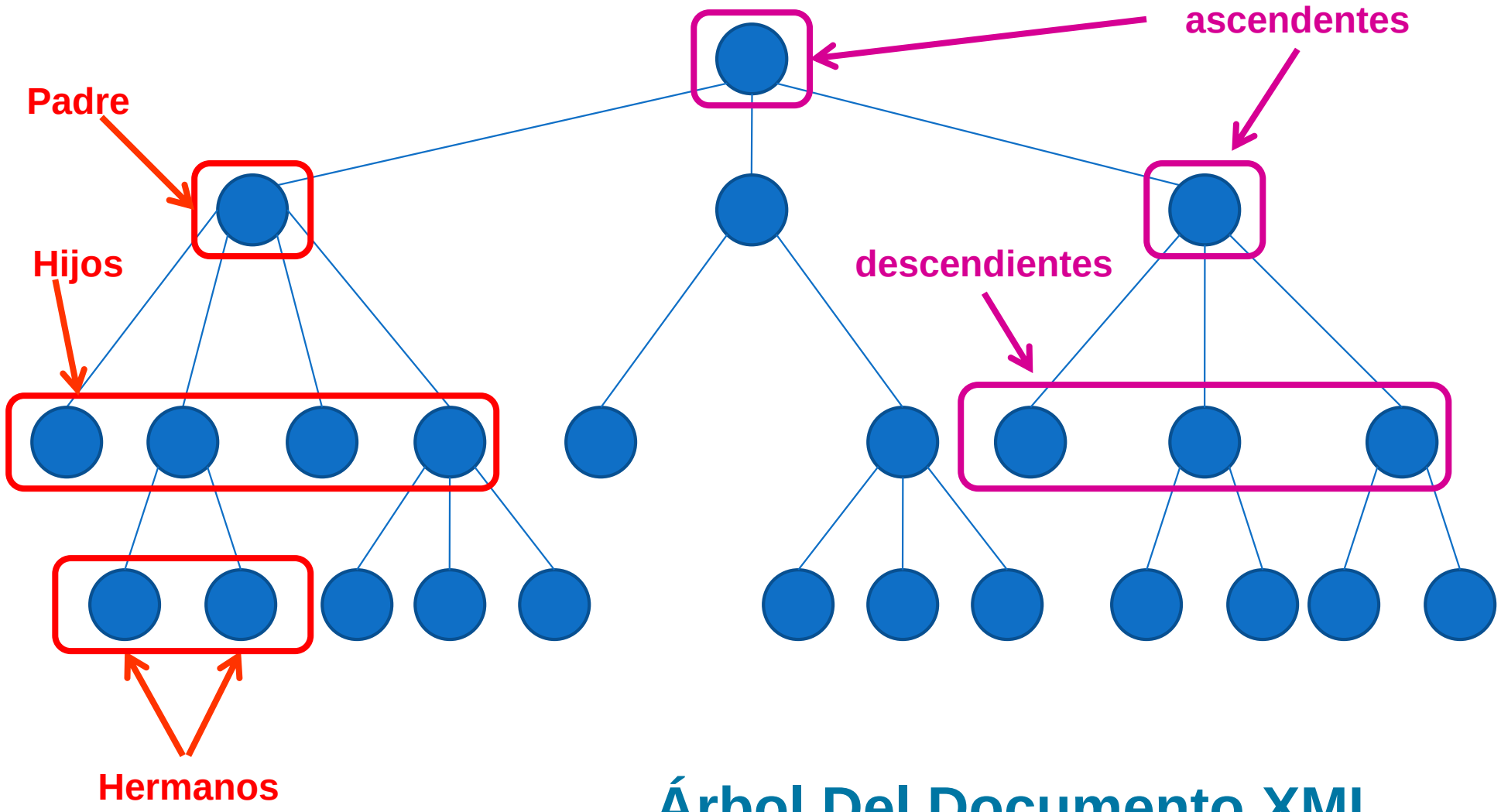
Usos XML

- Comunicación de datos entre aplicaciones, bases de datos
- Formatos de office: .docx, .pptx… son formatos xml, son zip de xml (añadir .zip, abrirlo y ver que está formado por un conjunto de archivos xml)
- Dibujo codificado con xml (abrir con XMLCopy Editor)

3. Estructura jerárquica de un documento XML

Estructura jerárquica de un documento XML

- Estructura de árbol



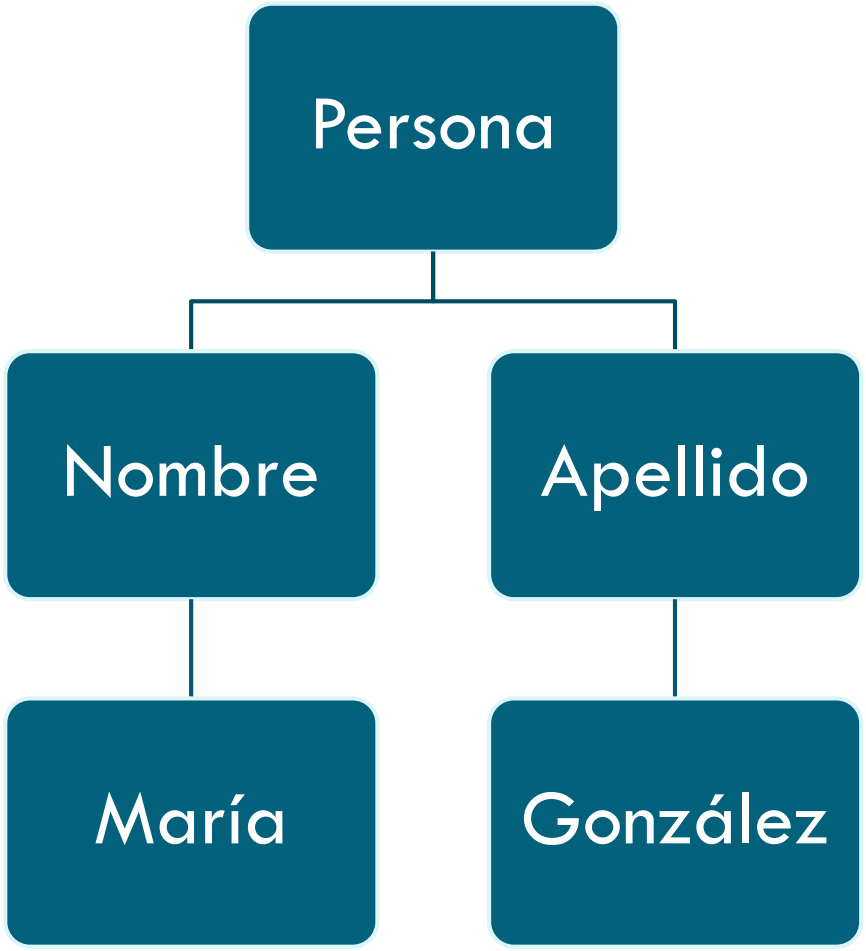
Árbol Del Documento XML

Ejemplo

- **Ejemplo:** El elemento **<persona>** es "padre" (contiene) a los elementos **<nombre>** y **<apellido>**, que son "hemanos" entre sí.

```
<persona>
        <nombre> María </nombre >
        <apellido> González </apellido>
</persona>
```

Ejemplo


Persona
Nombre
Apellido
María
González


Persona
Nombre
María
Apellidos
González

Visualización de un documento XML

- Para representar visualmente los datos de un documento XML:
  - Hoja de estilo CSS

```
<?xml-stylesheet type="text/css" href="estilos.css"?>
```

  - Hoja de transformaciones XSLT

```
<?xml-stylesheet  type="text/xsl"  href="transforma.xsl" ?>
```

  - Lenguaje de programación (Java o JavaScript)

4. Modelo de datos de un documento XML. Nodos

Tipos de componentes

1 **Raíz.**

- Se designa como /
- Punto de partida para recorrer el árbol

2 **Elementos.**

- Delimitadores/Contenedores de información
- Etiqueta de apertura y cierre

Tipos de componentes. Elementos

- **Elemento** raíz.
  - Todo documento XML bien formado debe contener un **único** elemento.
  - También se le llama **elemento documento**.
- **Elementos** sin contenido.
  - Puede tener **atributos.**
  - Una sola etiqueta.
    - → Elemento sin contenido y **sin atributos**.

```
<separador />
```

    - → Elemento sin contenido pero **con atributos**.

```
<separador cantidad= " 7" />
```

Tipos de componentes. Elementos

- Un **elemento** puede contener:
  - texto
  - atributos
  - otros elementos
  - o una mezcla de los anteriores

Tipos de componentes. Elementos

```
<libros>
  <libro categoria="niños">
    <titulo>Harry Potter</titulo>
    <autor>J K. Rowling</autor>
    <año>2005</año>
    <precio>29.99</precio>
  </libro>
  < libro   categoria ="web">
    <titulo>Learning XML</titulo>
    <autor>Erik T. Ray</autor>
    <año>2003</año>
    <precio>39.95</precio>
  </libro>
</libros>
```

<titulo>, <autor>, <año> y <precio> contienen texto

<libros> y <libro> tienen **contenido del** elemento, ya que contienen elementos.

<libro> tiene un **atributo** (categoría = "niños").

Tipos de componentes

3 Atributos

- Pares **nombre-valor**, con el signo igual.
- Datos adicionales de un elemento.
- En la etiqueta de apertura del elemento.
- Se tratarán como texto, entre **comillas simples o dobles**.
  - Almacenamiento de información sobre la información.

```
<distancia unidades="km">70</distancia>
```

  - Recogida de información distintiva entre elementos.

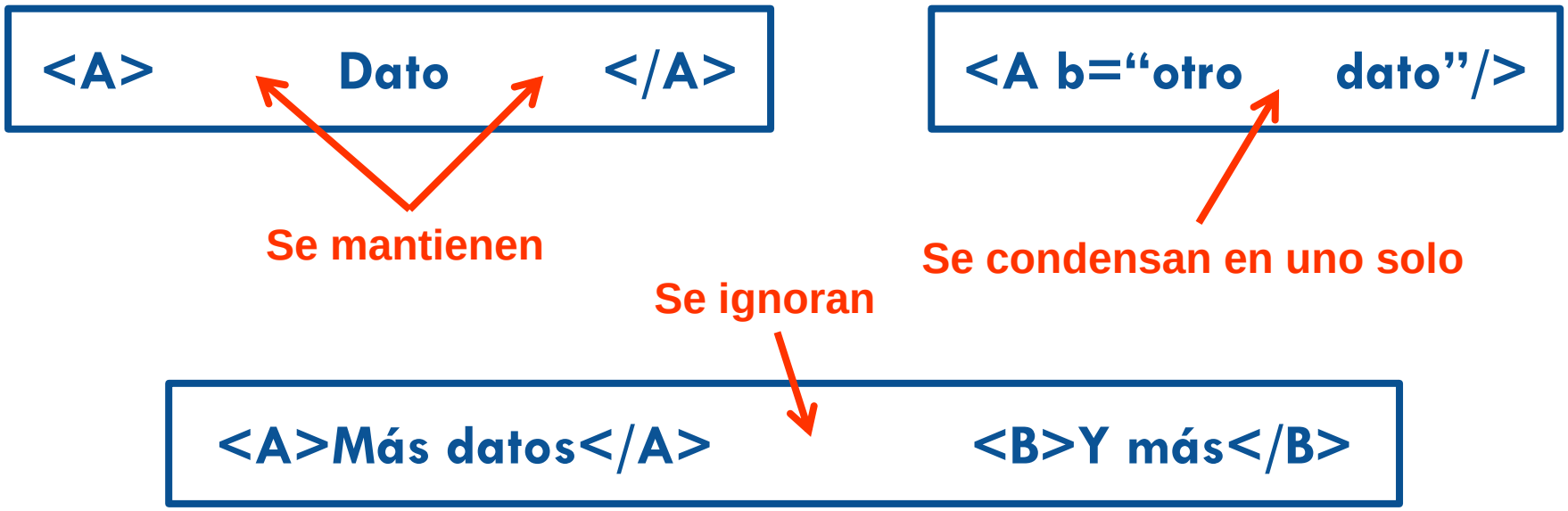
```
<persona nif="12345678Z">Alfonso Décimo El Sabio</persona>
```

Tipos de componentes

4 Texto

- Datos del documento XML
  - Contenido del elemento.
  - Valor del atributo.

Espacios	Representación
Tabulador	\t
Nueva línea	\n
Retorno de carro	\r
Espacio	\s



Tipos de componentes

5 Comentarios

- No dentro de una etiqueta de apertura y/o cierre

```
<!-- Comentario -->
```

6 Espacios

- Más adelante

7 Instrucciones de procesamiento

- Comienzan con **<?.....?>**
- Información para las aplicaciones

```
<?xml version="1.0" encoding="UTF-8"?>
```

Tipos de componentes

8 Entidades predefinidas

- Caracteres de marcado.

```
<bebida>Barnes &amp;amp; Noble</bebida>
```

Barnes & Noble

Entidad	Carácter
&amp;	&
&lt;	<
&gt;	>
&apos;	‘
&quot;	“

9 Secciones **CDATA**

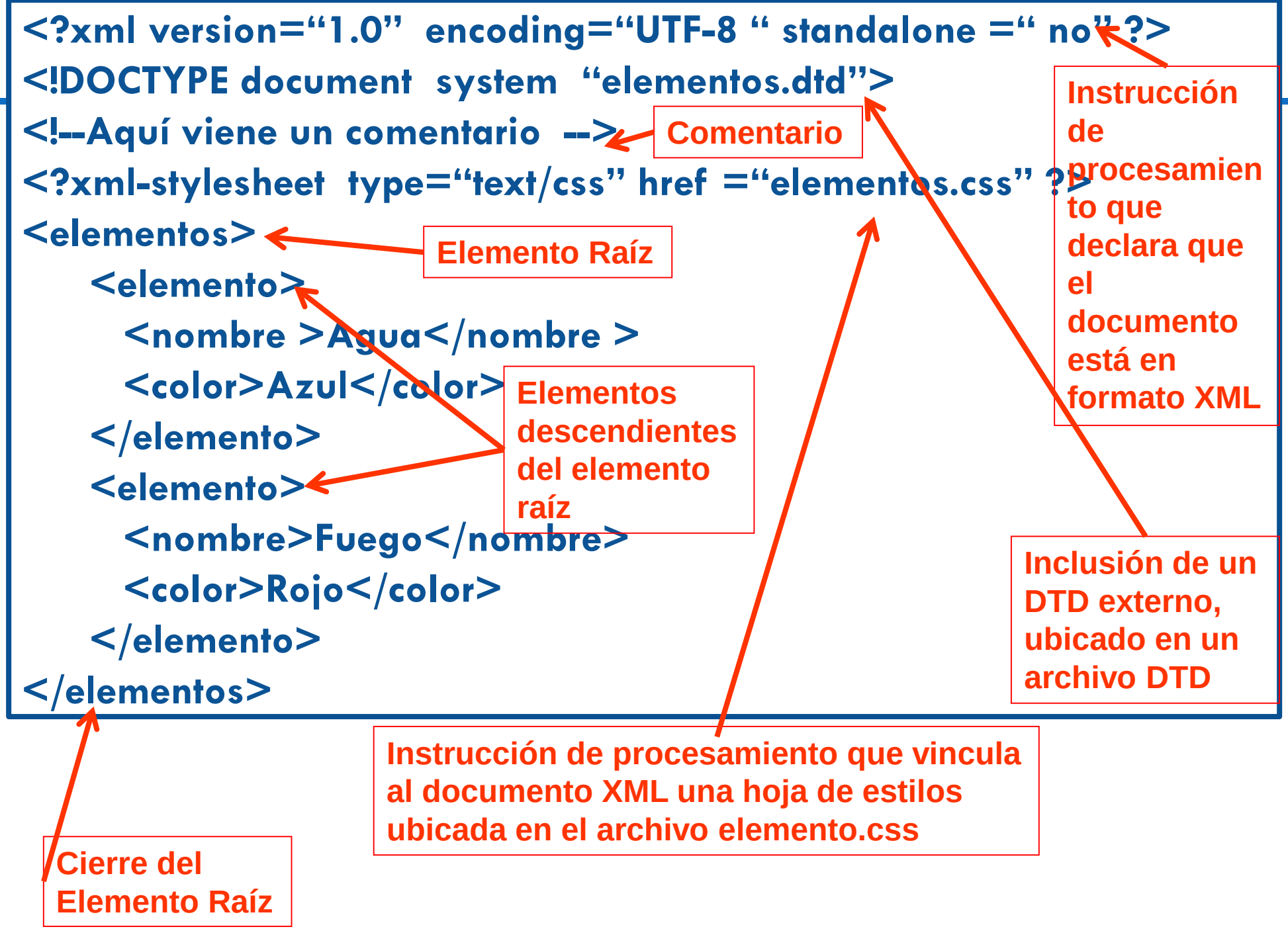
- Conjunto de caracteres que el procesador no debe analizar. Se puede usar < y > sin que se procese como tal.
- **No** antes del elemento

```
<codigo>
    <! [CDATA[
      <html><body><h3>Título</h3></body>
    ]]>
</codigo>
```

Tipos de componentes

10 Definición de tipo de documento DTD

- Define reglas sobre estructura de XML
- Documento XML bien definido asociado a un documento de declaración de tipos. +*adelante*.


<?xml version="1.0" encoding="UTF-8 " standalone =" no"?>
<!DOCTYPE document system "elementos.dtd">
<!--Aquí viene un comentario -->
<?xml-stylesheet type="text/css" href ="elementos.css" ??>
<elementos>
<elemento>
<nombre >Agua</nombre >
<color>Azul</color>
</elemento>
<elemento>
<nombre>Fuego</nombre>
<color>Rojo</color>
</elemento>
</elementos>
Comentario
Elemento Raíz
Elementos descendientes del elemento raíz
Instrucción de procesamiento que vincula al documento XML una hoja de estilos ubicada en el archivo elemento.css
Instrucción de procesamiento que declara que el documento está en formato XML
Inclusión de un DTD externo, ubicado en un archivo DTD
Cierre del Elemento Raíz

Nombres XML

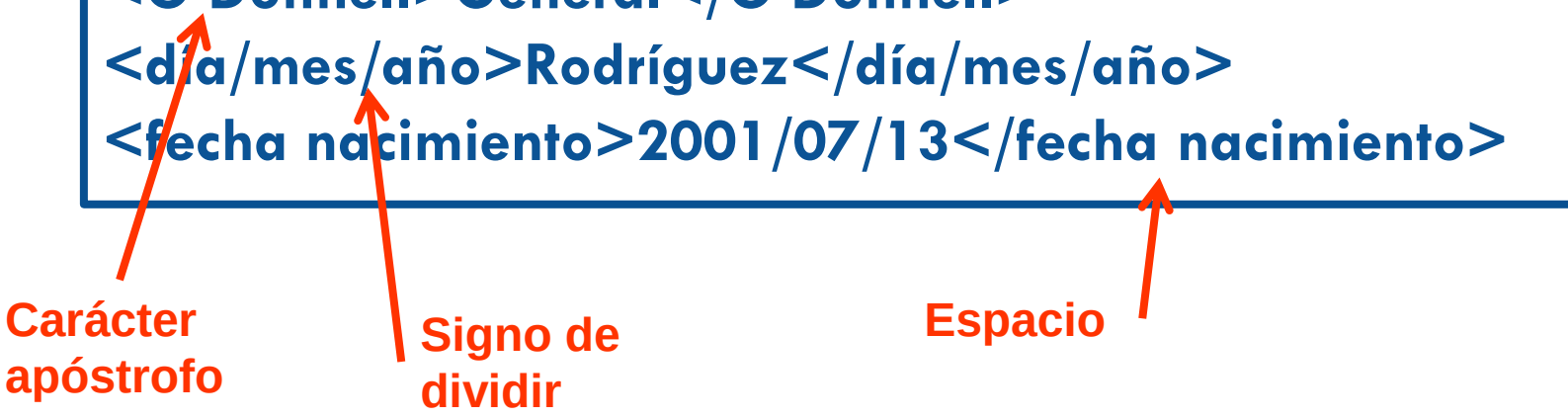
- **Reglas.**
  - Puede **empezar con** una letra (con o sin tilde), de un alfabeto no latino, subrayado o dos puntos (este último carácter es desaconsejado ya que se reserva su uso para los espacios de nombres).
  - Los **siguientes caracteres** pueden ser letras, dígitos, subrayados, guiones bajos, comas y dos puntos.
  - Los nombres que **empiezan por las letras XML**, en cualquier combinación de mayúsculas y minúsculas, se reservan para estandarización.
  - Distingue entre mayúsculas y minúsculas➔case-sensitive
  - No pueden contener:
    - Ningún carácter de espaciado.
    - Ningún otro carácter de puntuación de los ya citados como válidos. Esto incluye: comillas simples o dobles, signo de dólar, acento circunflejo, signo de porcentaje, punto y coma.

correcto

<Número_Seguridad_Social>50-12345678</Número_Seguridad_Social>
<primerApellido>Rodríguez</primerApellido>
<_cuenta_Tweeter>@follower</_cuenta_Tweeter>

errores

<O'Donnell>General</O'Donnell>
<día/mes/año>Rodríguez</día/mes/año>
<fecha nacimiento>2001/07/13</fecha nacimiento>



Buenas Prácticas de denominación

- Crear nombres descriptivos: <persona>, <apellido1>, <apellido2>.
- Crear nombres cortos y simples:
  - <titulo_libro>
  - No <el_titulo_del_libro>.
- Evitar "-". Puede interpretarse como resta.
- Evitar ".". Puede interpretarse como notación punto
- Evitar ":". Se reservan para los espacios de nombre (más tarde).
- Letras como éòá son perfectamente legales en XML, pero pueden surgir problemas si el software no es compatible con ellos.

Uso de elementos frente a uso de atributos

Los elementos	Los atributos:
• Representan **jerarquías** o contenido de unos dentro de otros. • Se pueden **extender** con otros elementos en su interior. • El **orden** en el que aparecen es representativo. • Pueden tener **atributos**. • Puede haber **múltiples ocurrencias** de un elemento.	• Van **asociados a los elementos**. • Son **modificadores** de la información. • Se suelen usar para **registrar metadatos**. • El **orden** en que aparecen dentro del elemento al que van asociados **no es representativo**. • **No se pueden extender** con otros elementos contenidos en su interior. • **No puede haber múltiples ocurrencias** de un atributo dentro de un mismo elemento. • **No son fácilmente ampliables** (cambios futuros)

Ejemplo. Tres opciones, mismo caso

```
<note date="2008-01-10">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

```
<note>
  <date>
    <year>2008</year>
    <month>01</month>
    <day>10</day>
  </date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

```
<note>
  <date>2008-01-10</date>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

Espacios de nombres

- Mecanismo para evitar conflictos de nombres
- Idénticos nombres pero diferentes definiciones.
- **Declaración** → como atributo

```
<nombre_elemento xmlns:prefijo="URI_del_espacio_de_nombres">
```

- **Uso**
  - Anteponer a elementos y atributos el prefijo asociado al espacio de nombres + carácter:

```
<info:pedido xmlns:info="empresa:espacios:info">
    <info:item info:id="i_13">Afeitadora eléctrica</info:item>
    …
</info:pedido>
```

Espacios de nombres

- Cualquier **cadena de texto** puede ser usada como **prefijo** del espacio de nombres.
- El **URI** del espacio de nombres sí debe ser único.
- **URI**: nombre lógico del espacio de nombres.
- **Ámbito de declaración = Espacio de nombres** → incluye los lugares donde se puede referenciar este espacio de nombres.
- Incluye el elemento donde se ha declarado y sus elementos descendientes (anteponer el prefijo del espacio de nombres).
- Se pueden declarar un espacio de nombres diferente para un elemento ascendiente y su descendiente.
- **xmlns = xml namespacing**

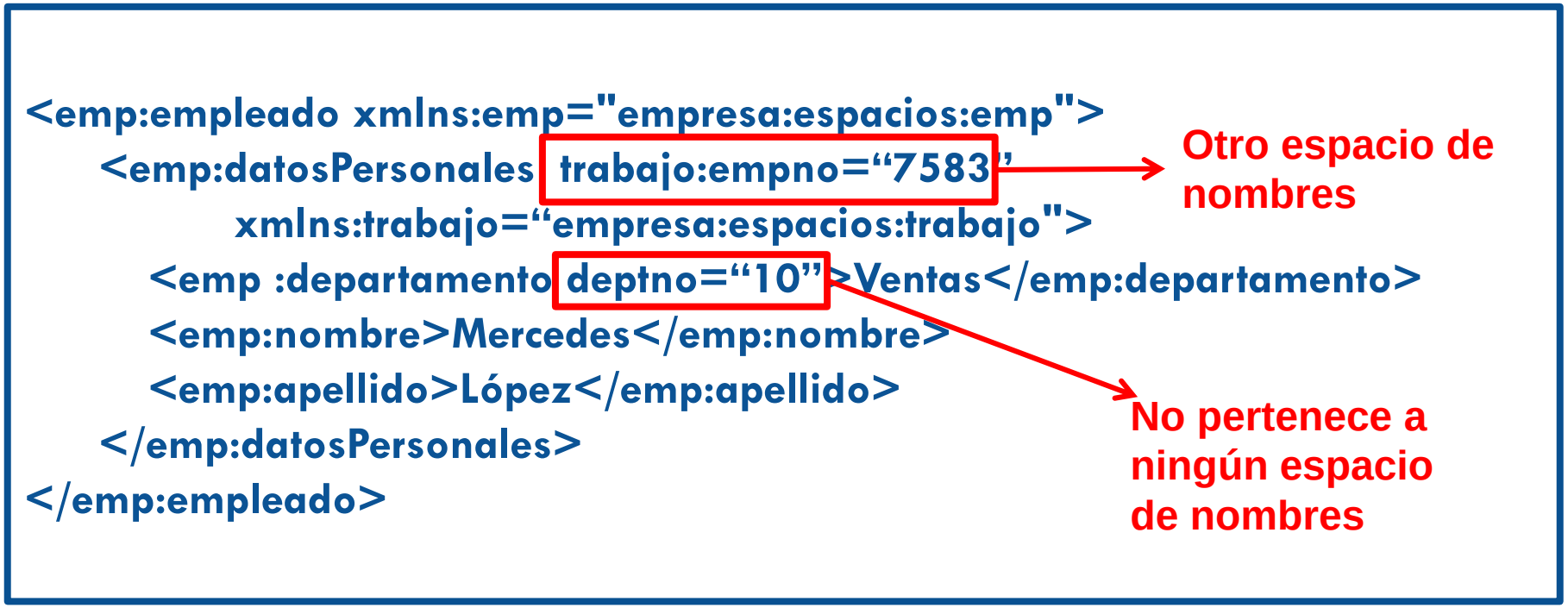
Espacios de nombres

- Dos **espacios de nombres**
  - empresa:espacios:dept
  - empresa:espacios:emp
- Dos **prefijos**
  - dept
  - emp

```
<dept:departamentos xmlns:dept="empresa:espacios:dept"
                    xmlns:emp="empresa:espacios:emp">
    <dept:departamento dept:deptno="10" >
      <emp:empleado emp:empno="7654">
        <emp:nombre>Roberto</emp:nombre>
        <emp:apellido>Mate</emp:apellido>
      </emp:empleado>
      <emp:empleado emp:empno="7891">
        <emp:nombre>Lucía</emp:nombre>
        <emp:apellido>Palmito</emp:apellido>
      </emp:empleado>
    </dept:departamento>
</dept:departamentos>
```

Espacios de nombres

- Los atributos pueden pertenecer al **mismo espacio** de nombres al que pertenece el elemento en el que están asociados o a otro diferente.
- Sin prefijo de espacio de nombres, el atributo no pertenecerá a ningún espacio de nombres.



Espacios de nombres

- Dos atributos con el mismo nombre pero con distinto prefijo son diferentes ➔ pueden estar asociados al mismo elemento.

```
<emp:empleado xmlns:emp="empresa:espacios:emp"
              xmlns:trabajo="empresa:espacios:trabajo">
  <emp:datosPersonales trabajo:empno="7583" emp:empno="e_7583">
    <emp:departamento deptno="10"> Ventas </emp:departamento>
    <emp:nombre>Mercedes</emp:nombre>
    <emp:apellido>López</emp:apellido>
  </emp:datosPersonales>
</emp:empleado>
```

Son diferentes.
Pueden asociarse
al mismo elemento

Espacio de nombres por defecto

- **No** se define un prefijo.
- **Ámbito de aplicación** → el del elemento en el que se ha declarado y sus elementos descendientes, pero no a sus atributos.
- **Ventaja**: evita añadir un espacio de nombres con prefijo a un documento XML grande ya creado, e ir añadiendo el prefijo a los elementos y atributos.
- **Inconveniente**: el espacio de nombres por defecto sólo afecta al elemento en el que está declarado y a sus descendientes, no a los atributos.

Espacio de nombres por defecto

- No afectará a **datosPersonales** ni al atributo **deptno="10"**

```
<empleado xmlns:emp="empresa:espacios:emp"
          xmlns:trabajo="empresa:espacios:trabajo">
  <datosPersonales  trabajo:empno="7583"  empno="e_7583">
    <departamento deptno="10"> Ventas </departamento>
    <nombre>Mercedes</nombre>
    <apellido>López</apellido>
  </datosPersonales>
</empleado>
```

Atributos especiales

- Hay algunos atributos especiales en XML:
  - **xml:space** ➔ le indica a la aplicación que usa el XML si los **espacios en blanco** del contenido textual de un elemento son significativos.
  - **xml:lang** ➔ permite especificar el **idioma** en el que está escrito el contenido textual de todo un documento o de sus elementos individuales. Los posibles valores son:
    - Un código de dos letras (ISO 639).
      - → **en** para English (**en-GB** y **en-US** ➔ inglés británico y americano).
    - Un identificador de idioma registrado por el **IANA** (Internet Assigned Numbers Authority - Autoridad de Números Asignados en Internet), que empiezan por el prefijo **i-** o **I-**.
    - Un identificador definido por el usuario. Empiezan por el prefijo **x-** o **X-**
  - **xml:base** ➔ permite definir una URL distinta a la del documento.

Parser XML

- **Analizador XML** → procesador que lee un documento XML y determina la estructura y propiedades de los datos en él contenidos.
- **Analizador estándar** →lee el documento XML y genera el árbol jerárquico asociado → permite ver los datos en un navegador o ser tratados por cualquier aplicación.
- **Analizador validador.**
  - Comprueba **Reglas de buena formación**
  - Comprueba **Valida** el documento contra un DTD o esquema.
  - Comprueba **Semántica** del documento
  - Informan de los errores existentes.
- Validadores XML en línea, como XML Validation (http://www.xmlvalidation.com).

Editores XML

- **XMLSpy**, de **Altova** (http://www.altova.com/es). Es un editor de
  - XML
  - XSLT
  - Xpath
  - XQuery...
  - Forma parte de MissionKit: StyleVision (diseño visual de hojas de transformaciones XSLT o XSL-FO y la generación de documentos de salida en formatos como PDF, HTML, RTF, PostScript...)
  - Representaciones gráficas de los distintos documentos editables

Editores XML

- **<oXygen/> de Syncro Soft** http://www.oxygenxml.com
  - Editor de tecnologías: XSD, XSLT o XQuery.
  - Interfaces de desarrollo gráfico.
  - Versiones para diversas plataformas (Windows, Linux, Mac...).
  - Módulos principales
    - XML Developer (el editor de XML en sí)
    - XML Author (generador de documentos en distintos formatos)

Editores XML

- **XML Copy Editor** http://xml-copy-editor.sourceforge.net
  - Software libre bajo licencia GNU GPL, siendo Gerald Schmidt el desarrollador principal.
  - Es un editor de XML, XSD, XSLT...
- **XMLPad Pro Edition** de **WMHelp** http://www.wmhelp.com/download.htm
  - Comprobación de buen formado de documentos XML
  - Validación frente a DTD o esquemas XML e inferencia de DTD o esquema XML a partir de un XML.
- **Exchanger XML Editor** http://www.exchangerxml.com
- **Liquid XML Editor** http://www.liquid-technologies.com/Xml-Studio.aspx

5. Corrección sintáctica: documento XML bien formado

Corrección sintáctica

- Especificación de XML = sintaxis.
  - Etiquetas.
  - Formato de la etiqueta.
  - Nombres aceptables elementos.
  - Colocación de atributos.
- **Bien formado** → cumplir reglas de W3C.

Reglas

- El documento puede empezar por una instrucción de procesamiento xml (**version** + opcionalmente **encoding** ➔ UTF-8 por defecto + **standalone** ➔ requiere o no archivos para procesarse)

```
<?xml  version="1.0"?>
<?xml version="1.0" encoding="UTF-8" standalone= "yes"?>
```

  - **Juego de caracteres**: colección de caracteres asociados cada uno a un número, llamado **punto de código**.
  - **Ejemplo** → Unicode, ISO Latin1(ISO-8859-1, 256 caracteres que coinciden con los 256 primeros de Unicode).
  - Una codificación (encoding) de caracteres determina cómo se representan en bytes los puntos de código.
  - **UTF-8** (Unicode Transformation Format de longitud variable de 8 bits) codificación de caracteres para el juego de caracteres Unicode.
  - **ASCII** es un juego de caracteres y una codificación de caracteres. Unicode es un superconjunto de ASCII (lo contiene).

Reglas

- Un único **elemento raíz,** que "cuelga" del nodo raíz (/).
- El elemento raíz tendrá como descendientes a todos los demás elementos.
- Los **elementos no vacíos** deben tener una etiqueta de apertura y otra de cierre.
- Los **elementos vacíos** deber cerrarse con />.
- Los elementos deben aparecer **correctamente anidados** en cuanto a su apertura y su cierre, no solaparse.
- Los nombres de elementos y atributos son **sensibles** a mayúsculas/minúsculas.
- Los valores de los atributos deben aparecer entre **comillas simples o dobles,** pero del mismo tipo.

Reglas

- No puede haber dos atributos con el mismo nombre asociados al mismo elemento.
- No se pueden introducir ni instrucciones de procesamiento ni comentarios en ningún lugar del interior de las etiquetas de apertura y cierre de los elementos.
- No puede haber nada antes de la instrucción de procesamiento **<?xml . . . ?>.**
- No puede haber texto antes ni después del elemento documento.
- No pueden aparecer los signos **<** ni **&** en el contenido textual de elementos ni atributos.

Actividad

Identifica qué documentos están bien formados y cuáles no (cada fila de la tabla pertenece a un documento diferente)
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<documento>Texto de prueba</documento>
< ?xml? > <documento>Texto de prueba</documento>
<?xml version="1.0"?> <DOCUMENTO/>
<?XML version="1.0"?> <Documento codigo="135"> <nombre>Artículo</nombre> <amplitud>Media</amplitud> </Documento>
<?xml version="1.0"?> <El documento> <nombre>Artículo</nombre> <amplitud>Media</amplitud> </El documento>

Actividad-SOLUCIÓN

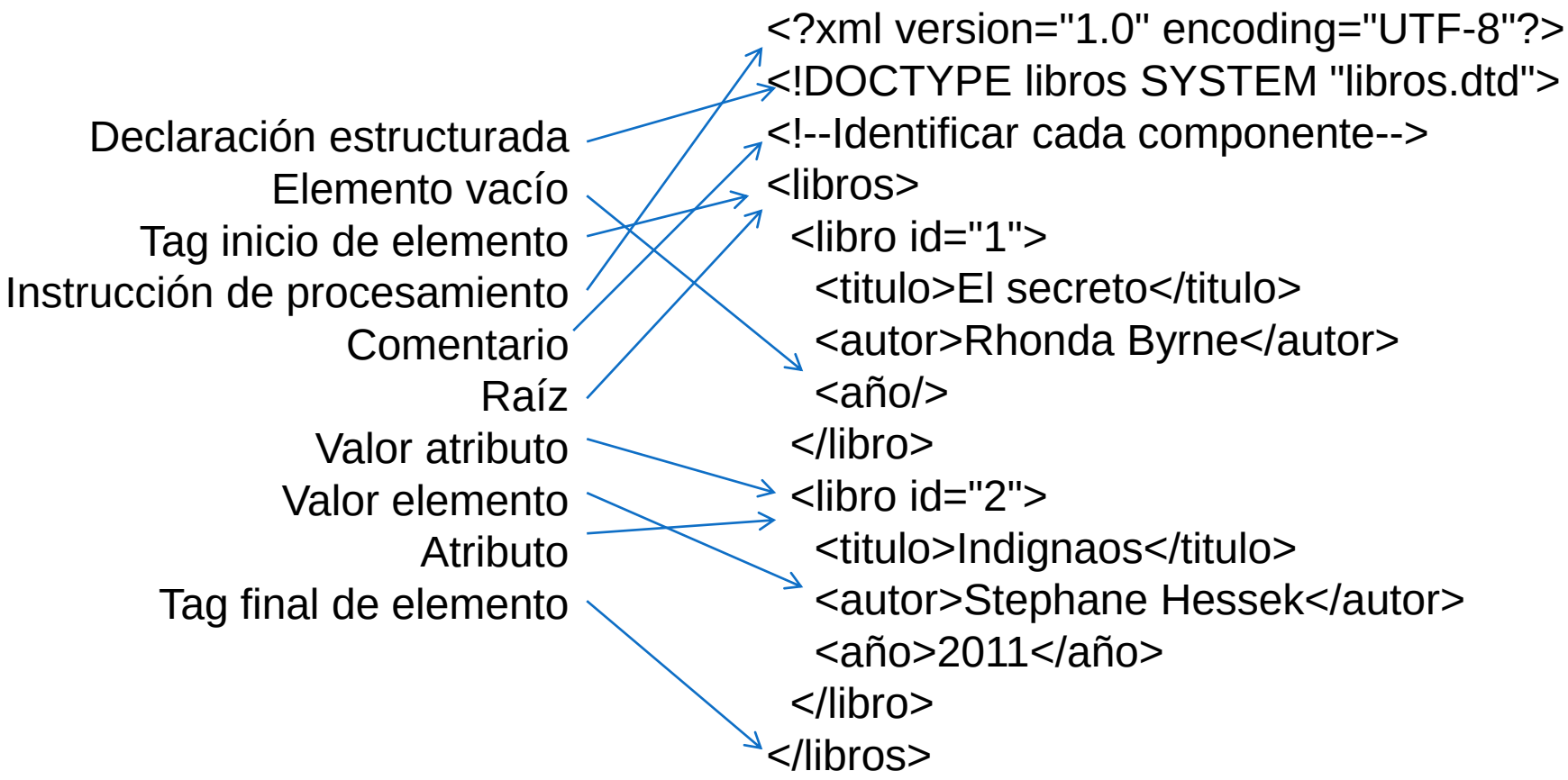
Identifica qué documentos están bien formados y cuáles no.	
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	✓
<documento>Texto de prueba</documento>	✓
< ?xml? > <documento>Texto de prueba</documento>	✗ La instrucción de procesamiento tiene como mínimo la versión
<?xml version="1.0"?> <DOCUMENTO/>	✓
<?XML version="1.0"?> <Documento codigo="135"> <nombre>Artículo</nombre> <amplitud>Media</amplitud> </Documento>	✗ La instrucción de procesamiento es con minúscula (xml)
<?xml version="1.0"?> <El documento> <nombre>Artículo</nombre> <amplitud>Media</amplitud> </El documento>	✗ Tag sin espacio

Ejercicio de emparejamiento

Declaración estructurada
Elemento vacío
Tag inicio de elemento
Instrucción de procesamiento
Comentario
Raíz
Valor atributo
Valor elemento
Atributo
Tag final de elemento

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE libros SYSTEM "libros.dtd">
<!--Identificar cada componente-->
<libros>
  <libro id="1">
    <titulo>El secreto</titulo>
    <autor>Rhonda Byrne</autor>
    <año/>
  </libro>
  <libro id="2">
    <titulo>Indiganaos</titulo>
    <autor>Stephane Hessek</autor>
    <año>2011</año>
  </libro>
</libros>
```

Ejercicio de emparejamiento



Ejercicio: errores en el documento

```
<?xml versión="1.0" ?>
<curso id="1">
   <nombre>Metodologías ágiles</nombre>
   <profesor>Ana</  profesor>
   <año></año>
 </curso>
 <CURSO id=5>
   <nombre>XML<Nombre/>
   <profesor>Pepe</profesor>
   <profesor>Rosa</profesor>
   <año>        2011        </año>
 </Curso>
   <curso id=2>
   <nombre>Bases de datos</titulo>
 </curso>
```

Ejercicio: errores en el documento

```
<?xml versión="1.0" ?>
<curso id=“1”>
   <nombre>Metodologías ágiles</nombre>
   <profesor>Ana</  profesor>
   <año></año>
 </curso>
 <CURSO id=5>
   <nombre>XML<Nombre/>
   <profesor>Pepe</profesor>
   <profesor>Rosa</profesor>
   <año>        2011        </año>
 </Curso>
   <curso id=2>
   <nombre>Bases de datos</titulo>
 </curso>
```

- Tilde en versión
- No existe elemento raíz (<centro>, p.e.)
- Faltan comillas en id=“5”
- Sobra el espacio en blanco en </ profesor>
- Elemento vacío es <año/>, pero se puede dejar como está.
- <CURSO> no puede estar en mayúsculas, distingue
- Idem con </Curso>
- Los espacios en el contenido 2011 no es erróneo pero puede dar problemas.
- Faltan comillas en id=“2”
- La etiqueta <Nombre/> va en mayúscula y con la barra al final.
- <nombre>….</titulo> etiquetasdiferentes