



# **XML EXTENSIBLE MARKUP LANGUAGE**

**Lenguajes de Marcas y Sistemas de Gestión de la Información**

**Profesor: Sergio Capdevila Diez**

# XML

- XML (*eXtensible Markup Language*, lenguaje de marcado extensible) es un lenguaje de marcado de propósito general
- Fue creado por el W3C a finales de los años 90. Se puede consultar este documento: <http://www.w3.org/TR/1998/REC-xml-19980210>
- Actúa como metalenguaje para definir otros lenguajes
  - Vocabularios XML
- Es una simplificación de SGML orientada a la web
- Estructura en forma de árbol
- En XML no existen marcas predefinidas como en HTML
  - Debe definirse un conjunto de marcas apropiado para la información que se quiere describir en cada caso
- Al igual que en HTML, las marcas se escriben entre los signos ‘<’ y ‘>’
  - Documento = contenido + marcas
- TEXTO PLANO, CASE SENSITIVE (p.e no es lo mismo <?xml version =“1.0”> q que <?XML version = “1.0”>

# Objetivos de XML

- XML se debe poder utilizar directamente en Internet
- XML debe admitir una gran variedad de aplicaciones
- XML debe ser compatible con SGML
- Debe ser fácil crear programas que procesen documentos XML
- El número de funcionalidades opcionales de XML deberá mantenerse en un mínimo absoluto, preferiblemente cero
- Los documentos XML deberán ser inteligibles para los humanos y razonablemente claros
- El diseño de XML deberá prepararse rápidamente
- El diseño de XML deberá ser formal y conciso
- Los documentos XML deberán ser fáciles de generar
- La concisión en las marcas XML tiene una importancia mínima

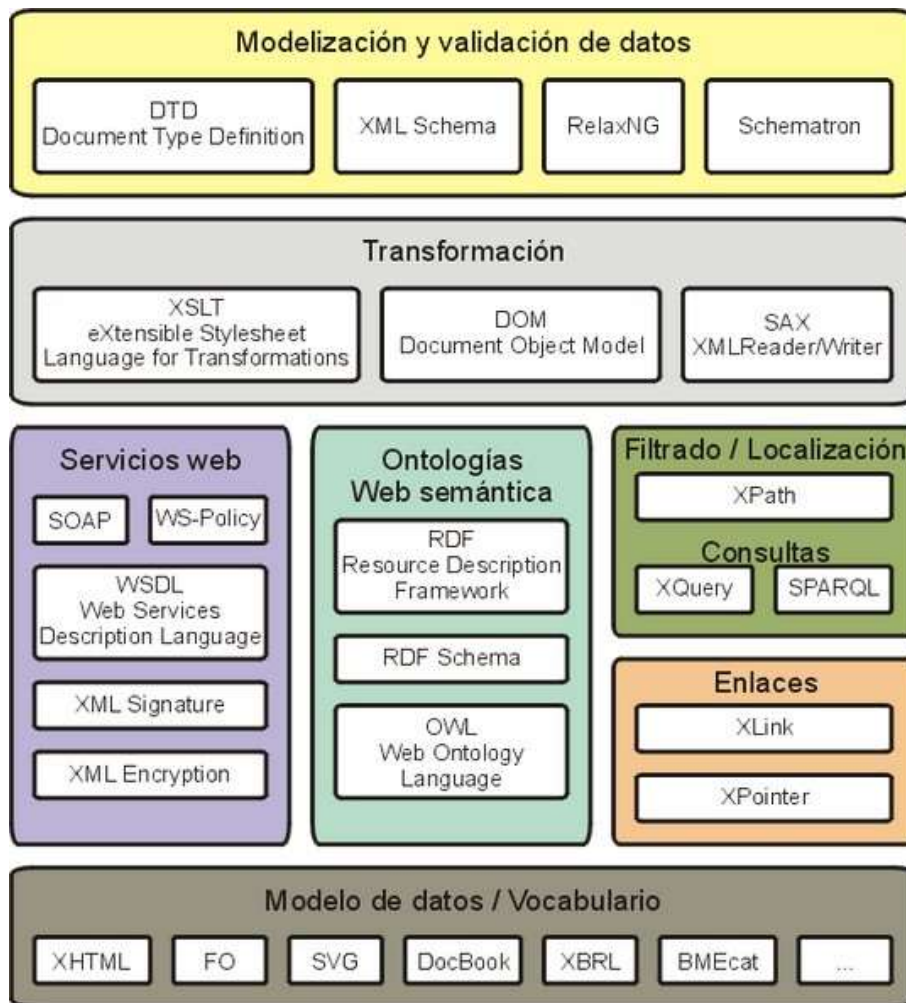
# Lo que NO es XML

- No es un lenguaje de programación, de manera que no existen compiladores de XML que generen ejecutables a partir de un documento XML.
- No es un protocolo de comunicación, así que no enviará datos por nosotros a través de Internet. (Los protocolos de comunicación como HTTP o FTP sí pueden enviar documentos con formato XML)
- No es un sistema gestor de bases de datos, aunque una base de datos puede contener datos de tipo XML y existan bases de datos nativas XML, que lo que almacenan son documentos XML.
- No es propietario, esto es, no pertenece a ninguna compañía.

# Otras recomendaciones XML

El W3C y otras organizaciones de normalización han publicado numerosas recomendaciones relacionadas con XML.

El cuadro siguiente cita algunas de ellas agrupándolas por temas:



# Ejemplo de documento XML

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Este es un ejemplo de documento XML -->
<!DOCTYPE anuncio SYSTEM "anuncioDTD.dtd">
```

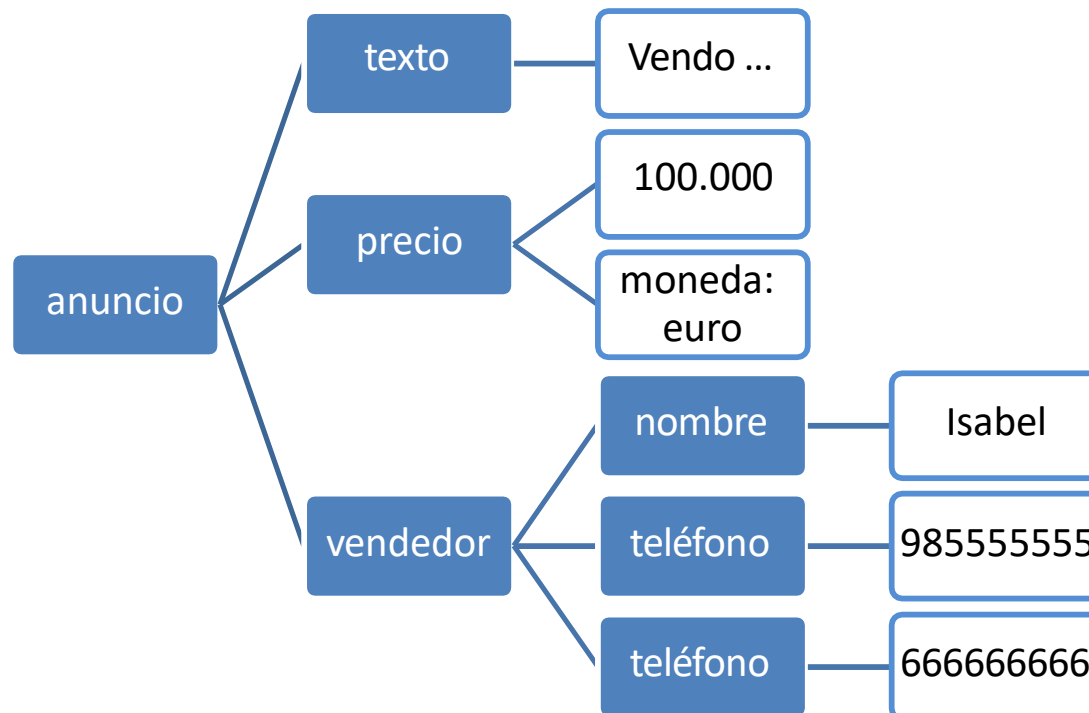
## Prólogo

```
<anuncio>
  <texto>vendo piso de 3 dormitorios en Gijón</texto>
  <precio moneda="euro">100000</precio>
  <vendedor>
    <nombre>Isabel</nombre>
    <telefono>985555555</telefono>
    <telefono>666666666</telefono>
  </vendedor>
</anuncio>
```

## Cuerpo

# Modelo de datos de un documento XML

- El documento se organiza jerárquicamente en forma de árbol
  - Los nodos internos son los elementos
  - Las hojas son los contenidos



# XML: ESTRUCTURA Y SINTAXIS: CONCEPTOS

- **Documento XML:** Un documento XML es un documento de texto plano (sin formato) que cumple las reglas de sintaxis de la recomendación W3C, se dice entonces que se trata de un documento XML "bien formado".
- **Procesador XML** (XML parser): Cuando una aplicación necesita leer un documento XML, la aplicación recurre a un procesador XML.
  - El procesador o intérprete XML lee el documento, analiza el contenido y le pasa la información en un formato estructurado a la aplicación.
- **Juego de caracteres:** Los documentos XML pueden estar codificados en distintos juegos de caracteres (ISO-8859-1, UTF-8, etc.).
- **Marcas y contenido:** El texto que contiene un documento XML se divide en marcas y contenido.
  - Las marcas pueden ser de dos tipos: etiquetas o referencias a entidades.
  - Todo lo que no son marcas es contenido.



# Componentes de un documento XML

- Prólogo
  - Declaración XML
  - Instrucciones de procesamiento
- Contenido - cuerpo
  - Elementos
  - Atributos
  - PCDATA
  - CDATA
  - Espacios en blanco
- Comentarios
- Notaciones
- Entidades

# Partes de un documento XML

- Un documento XML tiene dos partes
  - Prólogo
    - Es una zona de declaraciones
    - Es opcional, aunque es recomendado
- Elemento documento o elemento raíz
  - Es el elemento principal del documento, que contiene a todos los demás
  - Tiene que haber exactamente uno

# Prólogo

## ○ Declaración XML

- información de la versión de XML utilizada
- conjunto de caracteres utilizado para codificar la información
- ¿documento aislado? (standalone="yes|no")

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
```

## ○ Declaración del tipo de documento (opcional)

- Define el tipo y estructura del documento (sintaxis)

```
<!DOCTYPE anuncio SYSTEM "anuncioDTD.dtd">
```

# Prólogo

- Instrucciones de procesamiento (opcional)
  - Se marcan entre '<?' y '?>'
  - Proporcionan información que el procesador pasará a la aplicación XML
    - Dependiente del procesador del documento
    - P.e. Vincular una hoja de estilo a un documento

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

# Elementos

- Componentes básicos de un documento XML
- El identificador de los elementos debe comenzar con letra o subrayado bajo, después puede escribirse cualquier conjunto de caracteres excepto el espacio en blanco, los dos puntos, los símbolos de mayor y menor y la barra /
- Los elementos se delimitan mediante etiquetas formadas por el identificador
- Todos los elementos deben tener las etiquetas de inicio y de fin
  - `<identElemento>contenido del elemento</identElemento>`
- Los elementos vacíos tienen una forma abreviada
  - `<identificadorElementoVacio/>` (`<A/>` equivale a `<A></A>`)

# XML: DOCUMENTOS BIEN FORMADOS

- La sintaxis de un XML bien formado la define W3C:  
<https://www.w3resource.com/xml/well-formed.php>
- Un documento XML suele constar de un prólogo y un elemento raíz.
- El prólogo contiene información sobre el resto del documento.
  - En él se escriben las instrucciones de procesamiento, secciones CDATA, entidades, así como una descripción de la estructura del documento, generalmente recogida en una DTD (Document Type Definition) o en un Schema XML.
- Un documento XML debe estar **bien formado**, es decir debe cumplir las reglas de sintaxis de la recomendación W3C.

# Documentos bien formados

- Documentos que cumplen las reglas básicas de XML de modo que pueden ser procesados por un programa
- Se dice que un documento XML está **bien formado** cuando cumple las siguientes reglas:
  - El documento debe tener exactamente un elemento de nivel superior (elemento documento o elemento raíz)
    - La primera marca del documento es la marca del elemento raíz
  - Los elementos deben estar adecuadamente anidados, de modo que el contenido tenga una estructura de árbol
  - Cada elemento debe tener una marca de inicio y una marca de fin
    - Los elementos vacíos pueden indicarse con una marca especial
  - El nombre del tipo de elemento de una marca de inicio debe corresponder exactamente con su marca de fin correspondiente
    - En los nombres de los tipos de elementos se distingue entre mayúsculas y minúsculas
  - No puede aparecer un atributo más de una vez, en un mismo elemento
  - El valor de los atributos debe ir entre comillas

# EJERCICIOS

- INDICA SI CADA UNO DE LOS SIGUIENTES ES UN DOCUMENTO XML BIEN FORMADO. SI NO LO ES, EXPLICA POR QUÉ, ES DECIR QUÉ PUNTO DE LOS ANTERIORES NO CUMPLE, Y ESCRIBE UN DOCUMENTO CORRECTO:

A)

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

```
<persona>
```

```
    <nombre>Pedro</nombre>
```

```
    <DNI>123456789</DNI>
```

```
</persona>
```

```
<persona>
```

```
    <nombre>María</nombre>
```

```
    <DNI>987654321</DNI>
```

```
</persona>
```



# EJERCICIOS

**B)**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<texto>
```

Esto es un <negrita>texto</negrita> con formato.

```
</texto>
```

# EJERCICIOS

c)

<coche>

<marca>Peugeot</marca>

<modelo>205</modelo>

</coche>

# EJERCICIOS

D)

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

```
<amigos>
```

```
<amigo direccion="Avenida de la Felicidad"> Horacio Oliveira</amigo>
```

```
<amigo direccion="Calle de las Aceitunas"> Pilarín Cañete</amigo>
```

```
</amigos>
```

# EJERCICIOS

E)

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

```
<amigos>
```

```
<amigo telefono=999999999 telefono=666666666> Horacio  
Oliveira</amigo>
```

```
<amigo telefono=933333333>Pilarín Cañete</amigo>
```

```
</amigos>
```

# EJERCICIOS

F)

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

<ejercicio> Encuentra la solución de la inecuación:

<inecuacion> $2 \cdot x < 37$ <inecuacion>

</ejercicio>

# EJERCICIOS

G)

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

<ejercicio> Encuentra la solución de la inecuación:

<inecuacion> $2x < 37$ <inecuacion>

</ejercicio>

# EJERCICIOS

- Los siguientes documentos no están bien formados porque cada uno contiene dos errores (dos errores del mismo tipo cuentan como uno sólo). Corrige los errores y comprueba con XML Copy Editor que ya son documentos bien formados (si para corregir algún error hay que inventarse una etiqueta o atributo, utiliza un nombre que tenga relación con la información contenida en el documento).

# EJERCICIOS

H)

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

```
<pelicula>
```

```
<titulo>Con faldas y a lo loco</titulo>
```

```
<director>Billy Wilder</director>
```

```
</pelicula>
```

```
<pelicula>
```

```
<director>Leo McCarey</director>
```

```
<titulo>Sopa de ganso</titulo>
```

```
</pelicula>
```

```
<autor />barto</autor>
```



# EJERCICIOS

I)

```
<?xml version="1.0" encoding="UTF-8"?>
<deportistas>
  <deportista>
    <deporte Atletismo />
    <nombre>Jesse Owens</nombre>
  </deportista>
  <deporte Natación />
  <nombre>Mark Spitz</nombre>
</deportista>
</deportistas>
```

# EJERCICIOS

J)

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

```
<texto>
```

```
<Titulo>XML explicado a los niños</titulo>
```

```
<párrafo>El <abreviatura>XML</abreviatura>define cómo crear lenguajes de  
marcas.</párrafo>
```

```
<párrafo>Las marcas se añaden a un documento de texto para añadir  
información.</párrafo>
```

```
<http://>www.example.org</http://>
```

```
</texto>
```

# EJERCICIOS

K)

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

```
<geografia mundial>
```

```
<pais>
```

```
<pais>España</pais>
```

```
<continente>Europa</continente>
```

```
<capital>Madrid</capital>
```

```
</pais>
```

```
</geografia mundial>
```

# EJERCICIOS

L)

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

```
<programas>
```

```
<programa nombre="Firefox" licencia="GPL" licencia="MPL" />
```

```
<programa nombre="OpenOffice.org" licencia="LGPL" />
```

```
<programa nombre="Inkscape" licencia="GPL" />
```

```
</programas>
```

# EJERCICIOS

**M)**

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

```
<mundiales-de-futbol>
```

```
<mundial>
```

```
<1982>Mundial</1982>
```

```
</mundial>
```

```
</mundiales-de-futbol>
```

# EJERCICIOS

M)

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

```
<mundiales-de-futbol>
```

```
<mundial>
```

```
<1982>Mundial</1982>
```

```
</mundial>
```

```
</mundiales-de-futbol>
```

# EJERCICIOS

N)

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

```
<mundiales.de.futbol>
```

```
<mundial>
```

```
<pais="España" />subcampeona</pais>
```

```
<1982 />
```

```
</mundial>
```

```
</mundiales.de.futbol>
```

# EJERCICIOS

N)

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

```
<mundiales.de.futbol>
```

```
<mundial>
```

```
<pais="España" />subcampeona</pais>
```

```
<1982 />
```

```
</mundial>
```

```
</mundiales.de.futbol>
```



# EJERCICIOS

N)

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

```
<mundiales_de_futbøl>
```

```
<mundial>
```

```
<pais>españa</pais>
```

```
<1982 />
```

```
</mundial>
```

```
</mundiales_de_futbøl>
```

# EJERCICIOS

o)

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

```
<mundiales>
```

```
<mundial>
```

```
<pais>españa</pais>
```

```
<></>
```

```
</mundial>
```

```
</mundiales>
```

# EJERCICIOS

o)

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

```
<mundiales>
```

```
<mundial>
```

```
<pais>españa</pais>
```

```
<></>
```

```
</mundial>
```

```
</mundiales>
```

# EJERCICIOS

0)

```
<?xml version="1.0" encoding="UTF-8"?>
<mundiales.de.futbol>
  <mundial>
    <países>España&Italia</países>
    <continentes&subcontinentes>
      africa, america
    <continentes&subcontinentes>
  </mundial>
</mundiales.de.futbol>
```

# EJERCICIOS

0)

```
<?xml version="1.0" encoding="UTF-8"?>
<mundiales.de.futbol>
  <mundial>
    <países>España&Italia</países>
    <continentes&subcontinentes>
      africa, america
    <continentes&subcontinentes>
  </mundial>
</mundiales.de.futbol>
```

## Espacios de nombres: XML namespaces

- En ocasiones puede interesar utilizar en un mismo documento elementos y atributos pertenecientes a distintos vocabularios XML
- Esto puede conducir a un problema: la colisión de nombres
  - Podría suceder que elementos (o atributos) correspondientes a vocabularios diferentes tuviesen el mismo nombre
- Para evitar este problema se utilizan los espacios de nombres
  - La idea fundamental es cualificar el nombre de elementos y atributos con un prefijo correspondiente al espacio de nombres de su vocabulario, de forma que el nombre completo resultante sea único

# Espacios de nombres: XML namespaces

## ○ Declaración de espacios de nombres

- Se utiliza un atributo especial: `xmlns:prefijo`, que especifica el prefijo que precederá a los nombres de ese espacio de nombres
- Si no se indica ningún prefijo, `xmlns` declara el espacio de nombres predeterminado
- El atributo se puede definir en cualquier elemento. Define el uso de un espacio de nombres dentro de ese elemento y sus hijos
- El valor del atributo de `xmlns` debe ser una URI válida

```
<elemento xmlns:prefijo="URI">
```

```
<elemento xmlns="URI">
```

## ○ La URI que se especifica como valor del atributo `xmlns` sólo sirve como identificador único del espacio de nombres

- No referencia a ningún tipo de declaración, ni su contenido se utiliza para ningún tipo de validación

## Espacio de nombres

- EJ: DOCUMENTO XML1:

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

```
<banco>
```

```
    <color>marrón</color>
```

```
    <asientos>3</asientos>
```

```
</banco>
```

- EJ: DOCUMENTO XML2:

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

```
<banco>
```

```
    <nombre>BBVA</nombre>
```

```
    <dirección>Gran Via 1</dirección>
```

```
    <sucursales>2000</sucursales>
```

```
</banco>
```



## Espacio de nombres

- Para poder usarlos en un solo documento, crearíamos los espacios de nombres:

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

```
<b1:midoc
```

```
xmlns:b1="http://www.documentoXML1.com"
```

```
xmlns:b2="http://www.documentoXML2.com">
```



URIS (UNIVERSAL RESOURCE IDENTIFIER)  
EN ESTE CASO NO SON ENLACES SINO  
SOLO IDENTIFICADORES QUE DEBEN SER  
UNICOS EN EL DOCUMENTO

# Espacio de nombres

- Para poder usarlos en un solo documento, crearíamos los espacios de nombres:

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

```
<b1:midoc
```

```
  xmlns:b1="http://  
www.documentoX  
ML1.com"
```

```
  xmlns:b2="http://  
www.documentoX  
ML2.com">
```

```
<b1:banco>
```

```
  <b1:color>marrón</b1:color>
```

```
  <b1:asientos>3</b1:asientos>
```

```
</b1:banco>
```

```
<b2:banco>
```

```
  <b2:nombre>BBVA</b2:nombre>
```

```
  <b2:dirección>Gran Via 1</b2:dirección>
```

```
  <b2:sucursales>2000</b2:sucursales>
```

```
</b2:banco>
```

```
</b1:banco>
```

```
</b1:midoc>
```

# Ejemplos de espacios de nombres

```
<b:inversiones
  xmlns:b="http://www.bolsa.com"
  xmlns:g="http://www.geografia.es" >
  <g:pais g:nombre="Francia">
    <g:capital> Paris </g:capital>
    <b:capital> 1200 </b:capital>
  </g:pais>
</b:inversiones>
```

# EJERCICIOS

- EL OBJETIVO DE XML ES REEMPLAZAR A HTML (V/F)
- CUAL ES LA SINTAXIS CORRECTA DE LA DECLARACION QUE DEFINE LA VERSION DE XML?
- `<?XML version="1.0"?>`
- `<?xml version ="1.0"/>`
- `<?xml version ="1.0">`
- `<?xml version = "1.0"/?>`
- `<?xml version = "1.0"?>`

# EJERCICIOS

- COMENTA SI SON VERDADERAS O FALSAS LAS SGTES AFIRMACIONES:
  - A) TODOS LOS ELEMENTOS XML DEBEN ESTAR CORRECTAMENTE CERRADOS
  - B) TODOS LOS ELEMENTOS XML DEBEN ESTAR EN MINUSCULAS
  - C) TODAS LAS ETIQUETAS PUEDEN TENER ATRIBUTOS

# EJERCICIOS

- REPRESENTAR LA ESTRUCTURA EN ÁRBOL, Y DESPUÉS ESCRIBIR UN DOCUMENTO XML QUE REPRESENTA LA SIGUIENTE INFORMACIÓN SOBRE LA CARTA DEL MENÚ DE DESAYUNOS DE UN RESTAURANTE:

Nombre	Precio (en euros)	Descripción	Calorías
Gofres Belgas	5.95	Dos de nuestros famosos Gofres belgas con abundante sirope	650
Gofres Belgas con fresas	7.95	Ligeros gofres belgas cubiertos de fresas y nata montada	900
Gofres Belgas con frutas del bosque	8.95	Ligeros gofres belgas cubiertos con frutas del bosque y nata montada	900
Tostada Francesa	4.50	Dos gruesas rebanadas de nuestro pan francés casero	600
Desayuno de la casa	6.95	Dos huevos, bacon o salchicha, tostada y patatas fritas	950

# EJERCICIOS

## • SOLUCION

CARTA DESAYUNOS	{	DESAYUNO	{	NOMBRE
		DESAYUNO		PRECIO
		.....		DESCRIPCION
				CALORIAS
		DESAYUNO		

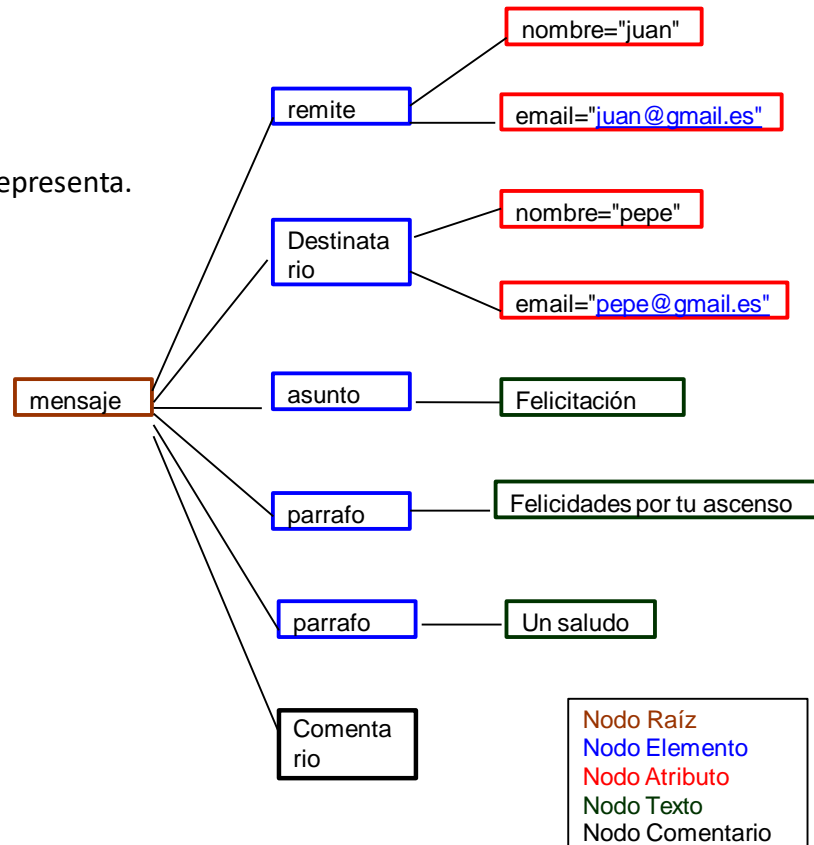
```
<?xml version="1.0" encoding="UTF-8"?>
<Carta desayunos>
  <desayuno>
    <nombre>Gofres Belgas</nombre>
    <precio>5.95</precio>
    <descripcion>Dos de nuestros famosos Gofres belgas </descripcion>
    <calorias>650</calorias>
  </desayuno>
  <desayuno>
    <nombre>Gobres Belgas con fresas</nombre>
    <precio>7.95</precio>
    <descripcion>Ligeros gofres cubiertos de fresas</descripcion>
    <calorias>900</calorias>
  </desayuno>
  ....
  ....
  <desayuno>
    <nombre>Desayuno de la casa</nombre>
    <precio>6.95</precio>
    <descripcion>Dos huevos, bacon o salchicha, tostada y patatas
fritas</descripcion>
    <calorias>950</calorias>
  </desayuno>
</Carta desayunos>
```



# EJERCICIO (I)

Dado el siguiente árbol de nodos.

SE PIDE: Escribir el documento XML que representa.





## EJERCICIO (II)

Dado el siguiente documento XML, crear el árbol correspondiente.

```
<?xml version="1.0"?>
```

```
<libro>
```

```
<titulo>El Ingenioso Hidalgo Don Quijote de la Mancha</titulo>
```

```
<autor>Miguel de Cervantes Saavedra</autor>
```

```
<fecha>
```

```
<publicacion>1605</publicacion>
```

```
<edicion>2009</edicion>
```

```
</fecha>
```

```
<localizacion>
```

```
<estanteria>B</estanteria>
```

```
<fila>7</fila>
```

```
</localizacion>
```

```
</libro>
```

## EJERCICIO (III)

DADO EL SIGUIENTE DOCUMENTO, SE PIDE: 1. ESTRUCTURA EN ÁRBOL CON SU CONTENIDO. 2. DOCUMENTO XML CORRESPONDIENTE. (NOTA: LOS CAMPOS NÚM, NÚMERO Y REF DEBEN REPRESENTARSE COMO ATRIBUTOS).

FACTURA Núm. 999			
CLIENTE Número: 879			
Nombre: José Rodríguez Teléfono: 91010101 0			
Datos Factura			
Ref.	Descripción	Cantidad	Precio
A101	Placa Base QDI	1	230
A105	DIMM DDR 512Mb	4	110
A107	Disco Duro 800Gb	1	150

## EJERCICIO IV

- DECIR SI EL SIGUIENTE XML ESTA BIEN FORMADO Y EN CASO DE NO ESTARLO CORREGIRLOS PARA QUE LO ESTE

```
<?xml version="1.0"?>
```

```
<documento>
```

```
  <p>Este <estilo subrayado=1>parrafo
```

```
XML</estilo></p>
```

```
  <p>empieza con el elemento <documento>&gt;</p>
```

```
  <p>Ahora pongo un elemento vacio</p>
```

```
  <imagen fichero="imagen.gif">
```

```
</documento>
```

## EJERCICIO IV

- CORREGIR EL XML PARA QUE MUESTRE

```
<?xml version="1.0"?>
<documento>

  <p>Este <estilo subrayado=1>parrafo
  XML</estilo></p>
  <p>empieza con el elemento <documento>&gt;</p>
  <p>Ahora pongo un elemento vacio</p>
  <imagen fichero="imagen.gif">

</documento>
```

```
<del><documento>
  <del><p>
    Este
    <estilo subrayado="1">parrafo XML</estilo>
  </del><p>
    <p>empieza con el elemento <documento></del></p>
    <p>Ahora pongo un elemento vacio</del></p>
    <imagen fichero="imagen.gif"/>
  </del><documento>
```

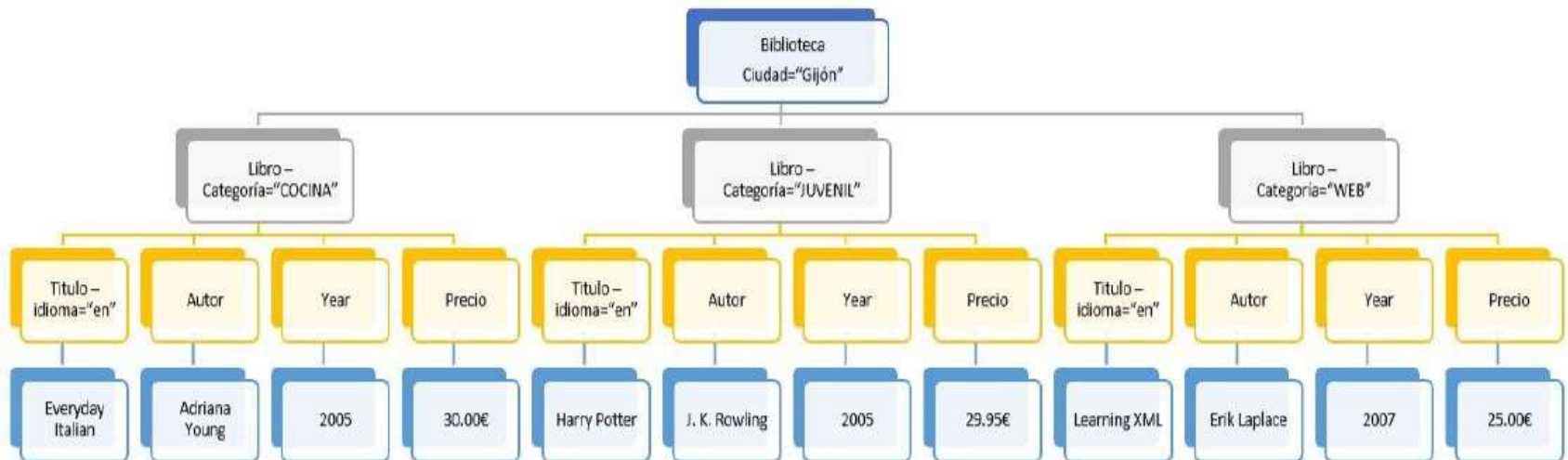
## EJERCICIO V

- DECIR SI EL SIGUIENTE XML ES CORRECTO O NO Y CORREGIRLO EN SU CASO:

```
<?xml version="1.0"?>
<juegos>
  <juego id="metal gear"/>
  <titulo>Metal Gear Solid V</titulo>
  <autor nombre=Hideo nombre=Kojima>
  <descripción breve>Es >>>>> Metal Gear Solid 2</descripción breve>
</juego>
</juegos>
```

EJERCICIO VI: Dado el siguiente árbol, guardar los datos como un fichero xml bien formado.

- Ciudad, categoría e idioma serán atributos.
- La última fila muestra los valores de los elementos.



## EJERCICIO VII

- ¿ESTA BIEN FORMADO EL SIGUIENTE XML?

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

```
<banco>
```

```
  <color>marrón</color>
```

```
  <banco>
```

```
    <cajero>automatico</cajero>
```

```
    <horario>tardes</horario>
```

```
  </banco>
```

```
  <asientos>3</asientos>
```

```
</banco>
```

# Bibliografía

- W3C: <http://www.w3.org/TR/1998/REC-xml-19980210>
- Curso "XML: Lenguaje de Marcas Extensible"  
Autor: Bartolomé Sintés Marco
- Libro: Lenguajes de Marcas y Sistemas de Gestión de la Información  
Autor: J.M. Castro Ramos  
Editorial: Garceta
- Introducción a XML  
Autor: Alfredo Reino