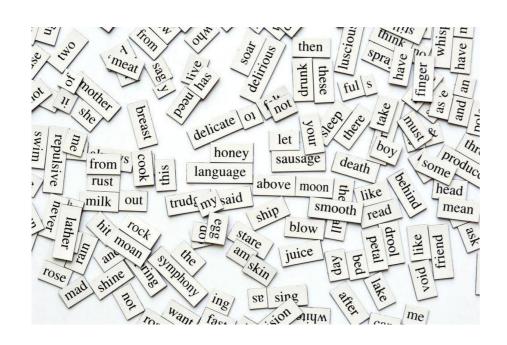


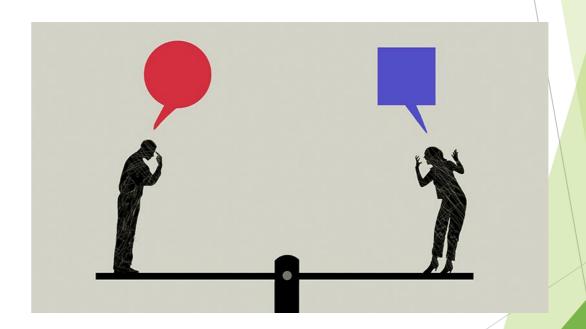
UD1 - INTRODUCCIÓN A LOS LENGUAJES DE MARCAS. XML

### UD 1 - INTRODUCCIÓN A LOS LLMM. XML Índice de Contenidos

- ▶ 1 ¿Qué es un lenguaje?
- > 2 Concepto de un lenguaje de marcas
- > 3 El documento de un LM
- 4 Ventajas de usar documentos de LLMM
- ▶ 5 Características de los LLMM
- ► 6 Ámbitos de aplicación
- > 7 Clasificación de los LLMM
- ▶ 8 XML

- 1 ¿Qué es un lenguaje?
- ▶1 ¿Qué es un lenguaje?





Es un **sistema de signos** que utiliza una comunidad para comunicarse oralmente o por escrito.

> ¿ Qué es un **signo**?

Es un objeto, hecho o fenómeno que representa o evoca a otro objeto, hecho o fenómeno



Los iconos de una aplicación son SIGNOS, representan a otro objeto

La Lengua estudia los lenguajes en varios niveles:

#### Los niveles de la lengua.

Nivel	Objeto de estudio	Ciencia
Fónico	Estudia los sonidos, fonemas, el acento y la entonación.	Fonología Fonética
Morfológico	Estudia la estructura interna de las palabras: sus partes, los morfemas y sus clases.	Morfología
Sintáctico	Se ocupa de las funciones de las palabras en sintagmas y dentro de oraciones.	Sintaxis
Semántico	Posee un doble objeto de estudio: la significación de las palabras y la significación de las oraciones.	Semántica



- Frases con fallos a nivel sintáctico o semántico:
  - Rotos libros están los.
  - La coche va muy rápida
  - La luz viajas por el espacio
  - El melón traduce la pared
  - La cabra llueve de noche
- ► Frase ambigua:
- Lourdes no quiere a su tía porque es muy envidiosa.

¿Quién es envidiosa? ¿Lourdes o su tía?

- Este año vais a aprender muchos lenguajes informáticos y tendréis que poner atención para:
- Escribir bien → Conocer bien la sintaxis del lenguaje
- Expresarte con claridad, evitando ambigüedades → Conocer bien la semántica del lenguaje





# UD 1 - INTRODUCCIÓN A LOS LLMM. XML 2 - Concepto de un lenguaje de marcas

▶ 2 – Concepto de un lenguaje de marcas

Es un lenguaje que utiliza etiquetas o marcas textuales como signos. Estas signos acompañan a otra información, aportándole un nuevo significado.

Supongamos que tenemos la siguiente información:

Programación

 $\mathbf{0}$ 

¿ Qué queremos decir?

Juan Sin Miedo



# UD 1 - INTRODUCCIÓN A LOS LLMM. XML 2 - Concepto de un lenguaje de marcas

- ► Si le añadimos las siguientes marcas todo tiene más sentido:
  - <asignatura>Programación</asignatura>
  - <calificación>0</calificación>
  - <alumno>Juan Sin Miedo</alumno>

- En esto consiste el juego, veamos otro ejemplo:
  - <Negrita>Esto se escribirá en letra negrita</Negrita>
  - <Cursiva>Esto se escribirá en letra cursiva</Cursiva>
  - <Negrita><Cursiva>Esto se escribirá en letra cursiva y negrita</Cursiva></Negrita>

# UD 1 - INTRODUCCIÓN A LOS LLMM. XML 2 - Concepto de un lenguaje de marcas

</curso>

Además unas etiquetas pueden contener a otras:

```
<curso>
    <etapa>Ciclo Formativo de grado Superior</etapa>
    <especialidad>Desarrollo de Aplicaciones Web</especialidad>
    <nivel>10</nivel>
    <alumnos>
    <alumno>Borja Mon DeYork</alumno>
    <alumno>Armando Adistancia</alumno>
    <alumno>Josechu Letón</alumno>
    </alumnos>
```

#### 3 - El documento de un LM

#### ▶ 3 – El documento de LM

Normalmente se presenta como un fichero de texto plano que consta de:

- Cabecera o prólogo: indica el tipo de lenguaje de marcas utilizado (y opcionalmente otra información)
- Cuerpo: contiene al menos un elemento.

```
<?xml version="1.0"?>
<Catalog>
   <Book id="bk101">
      <a href="#"><Author>Garghentini, Davide</author></a>
      <Title>XML Developer's Guide</Title>
      <Genre>Computer</Genre>
      <Price>44.95</Price>
      <PublishDate>2000-10-01</PublishDate>
      <Description>An in-depth look at creating
      applications with XML.</Description>
   </Book>
   <Book id="bk102">
      <Author>Garcia, Debra</Author>
      <Title>Midnight Rain</Title>
      <Genre>Fantasy</Genre>
      <Price>5.95</Price>
      <PublishDate>2000-12-16</PublishDate>
      <Description>A former architect battles
      corporate zombies and evil sorceress
      </Description>
   </Book>
 /Catalog>
```

## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 3 - El documento de un LM

Entendemos por elemento de un documento el siguiente conjunto:

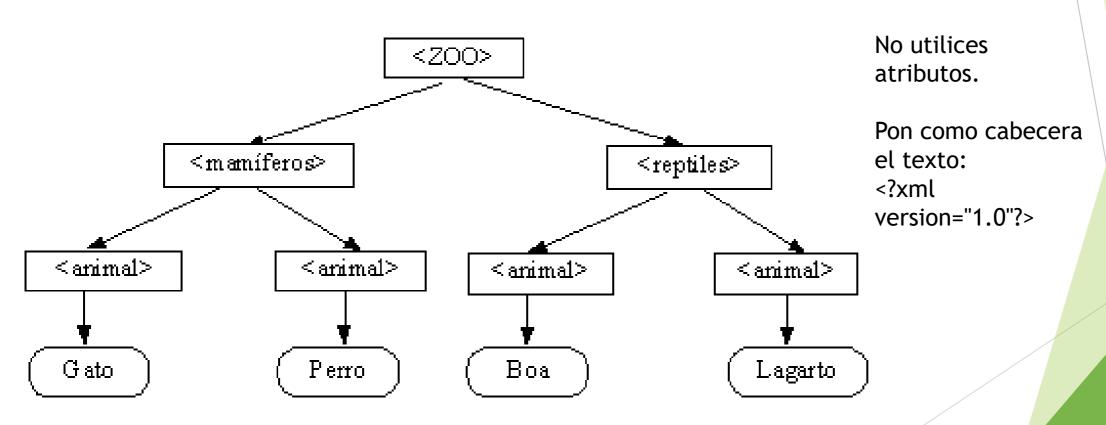
<etiqueta>información</etiqueta>

Además, la etiqueta de apertura puede opcionalmente llevar uno o más atributos:

Los atributos son pares nombre="valor" que sirven para aportar una información de detalle extra a la etiqueta que lo contiene. 12

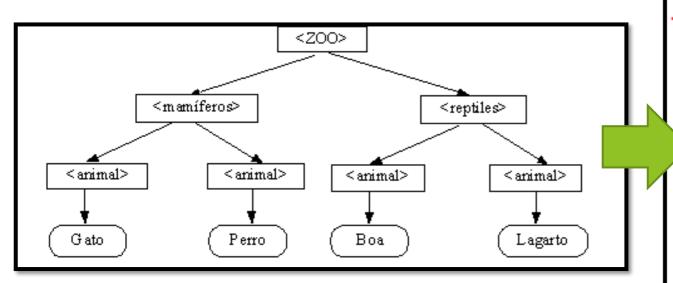
## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 3 - El documento de un LM

Abre el 'Bloc de notas' y teclea el documento de lenguaje de marcas asociado al siguiente árbol:



## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 3 - El documento de LM

Por último, sálvalo como "zoo.xml" y ábrelo con un navegador



```
<?xml version="1.0"?>
<z00>
  <mamiferos>
       <animal>Gato</animal>
       <animal>Perro</animal>
   </mamiferos>
  <reptiles>
       <animal>Boa</animal>
       <animal>Lagarto</animal>
   </reptiles>
</zoo>
```

# UD 1 - INTRODUCCIÓN A LOS LLMM. XML 4 - Ventajas de usar documentos de LLMM

#### 4 – Ventajas de usar documentos de LLMM

- Permiten organizar y estructurar información de cualquier ámbito.
- Permiten un tratamiento eficiente de la información gracias a la estructura en árbol de los documentos.
- Permiten intercambiar información entre dos máquinas cualesquiera (Windows, Linux, Mac...)
- Permiten separar el formato de presentación del contenido.
- Permiten añadir una semántica nueva a cualquier contenido.
- Son legibles por el "humano medio".

## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 5 - Características de los LLMM

#### 5 – Características de los LLMM

- Los documentos de LLMM usan el texto plano para codificar tanto las marcas como la información.
- Son compactos, ya que las etiquetas se entremezclan con el contenido en el mismo documento.
- Son fáciles de procesar por un ordenador.
- Son flexibles, pudiéndose adaptar a contextos muy diversos (continúa)

### UD 1 - INTRODUCCIÓN A LOS LLMM. XML 5 - Características de los LLMM

- Siguen una sintaxis definida (reglas para escribir correctamente las etiquetas de un documento)
- Siguen una semántica definida (reglas que indican cómo se combina correctamente las etiquetas de un documento)



# UD 1 - INTRODUCCIÓN A LOS LLMM. XML 6 - Ámbitos de aplicación

- ▶ 6 Ámbitos de aplicación
- Creación de documentos (libros, presentación, manuales).
- Tecnologías de internet (web, sindicación, interfaz de usuario, servicios).
- Lenguajes especializados diversas disciplinas: gráficos 2D y 3D, matemáticas, arquitectura, música, aeronáutica, procesamiento de voz, multimedia, mensajería instantánea, etc.

## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 7 - Clasificación de los LLMM

- 7 Clasificación de los LLMM
- Según la finalidad con que se crearon distinguimos dos grandes grupos:
  - LLMM de presentación: son aquellos cuyas etiquetas indican cómo debe presentarse un documento (formato, tipo de fuente, tamaño...)
  - LLMM descriptivos o semánticos: son aquellos cuyas etiquetas aportan un significado nuevo al contenido describiéndolo de algún modo.



## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 7 - Clasificación de los LLMM

- A su vez los LLMM de presentación se pueden clasificar en:
- Procedimentales: además de las etiquetas de formato existen otras destinadas a procesar el texto permitiendo añadir gráficos, fórmulas, numeración de página automática, formato global...
  - Ejemplos: Tex, LaTex
  - Son lenguajes que se deben "compilar" para obtener el documento final. Visitar: <a href="https://papeeria.com/">https://papeeria.com/</a> -> Live demo
- No procedimentales: no tienen etiquetas de procesado.
  - ► Ejemplos: HTML, RTF

## UD 1 - INTRODUCCIÓN A LOS LLMM. XML 7 - Clasificación de los LLMM

- Los LLMM descriptivos o semánticos más conocidos a su vez se podrían clasificar según quién es el "padre" de quién:
- GML (Generalized Markup Language 1970)
  - ► SGML (Standard GML 1986)
    - XML (Extensible Markup Language 1998)
      - ▶ Dialectos del XML: RSS, Atom, CSS...
- Para saber más:

https://es.wikipedia.org/wiki/Lenguaje\_de\_marcado

- 8 XML (Extensible Markup Language)
  - ► Es un LM semántico, fácil de aprender y que permite expresar y transmitir cualquier información.
  - Las etiquetas XML no están predefinidas, por eso es extensible.
  - Es un metalenguaje, es decir un lenguaje que permite crear otros lenguajes.
  - Es la base de muchos lenguajes de la web.



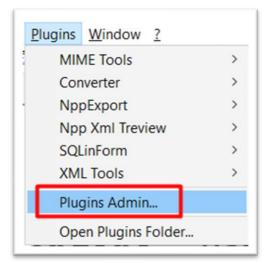
- ▶ 8.1 Herramientas para trabajar con XML
- Durante el curso vamos a utilizar:
- Editores XML: son editores de texto plano pero con funcionalidades extra para ayudarnos a escribir documentos XML
- Analizadores XML: son programas que verifican si un documento es correcto o no.
- Bases de datos XML: son BBDD que almacenan la información en XML y permiten la consulta de los datos almacenados

- Durante este tema usaremos:
- Notepad++: es un editor de texto plano multipropósito de software libre y que puede ser ampliado mediante plugins.
  - https://notepad-plus-plus.org/downloads/

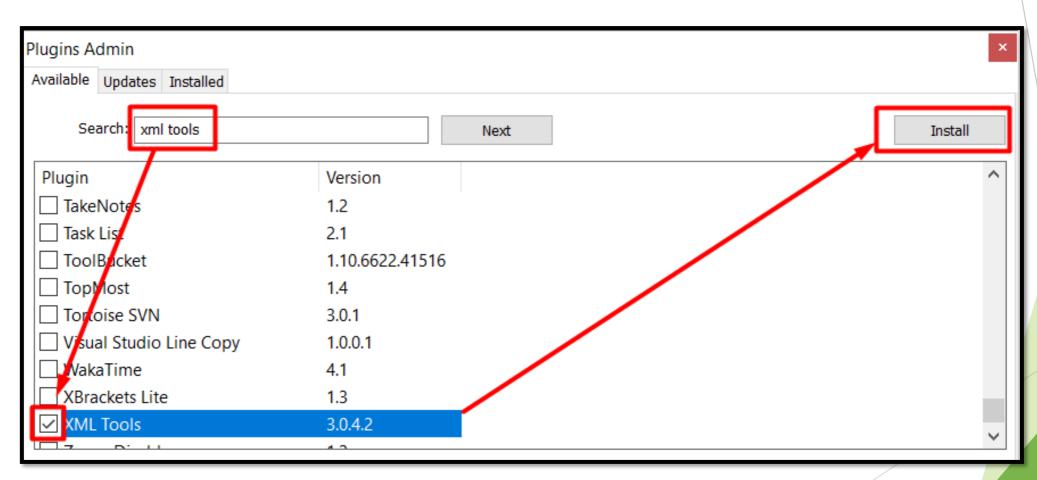
 Plugin XML Tools para Notepad++: nos ayudará en la escritura de documentos XML y nos permitirá analizar

los documentos escritos.

Para instalarlo hacemos desde el Notepad++:



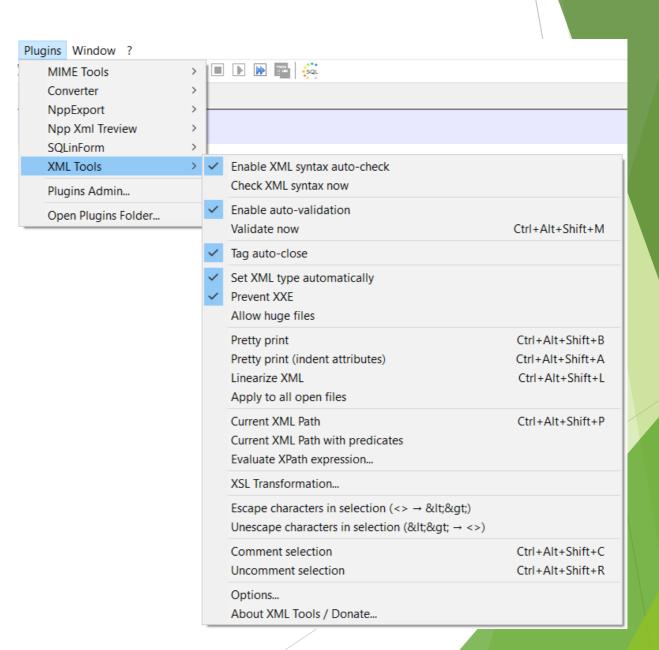
A continuación, lo buscamos y lo instalamos:



#### 8 - XML

 Una vez instalado podemos observar las distintas opciones desde el menú: Plugins -> XML Tools

A medida que nos vayan haciendo falta iremos explicando las distintas opciones.



- También usaremos:
- Cualquier navegador como analizador XML: aunque el plugin anterior es capaz de analizar documentos XML también utilizaremos cualquier navegador del sistema para ello.



#### ▶ 8.2 – Documentos XML bien formados y/o validados

Cuando un documento XML bien formado es aquel que cumple todas las reglas sintácticas del lenguaje.

En este tema estudiaremos estas reglas sintácticas y aprenderemos a escribir este tipo de documentos.

Cuando un documento XML bien formado, además, cumple con un conjunto de reglas gramaticales entonces se dice que es un documento XML válido o validado.

Este tipo de documentos los estudiaremos en la unidad didáctica de "Validación de XML"



- ▶ 8.3 Sintaxis XML
- Para crear correctamente documentos XML tenemos que conocer unas cuantas reglas de sintaxis:
- Los documentos se componen de:
  - Una cabecera o prólogo opcional, que debe aparece en primer lugar.
  - Un cuerpo que tiene al menos un elemento.

```
Cabecera 

<
```

- La cabecera no sigue las reglas de etiquetado de los elementos XML precisamente para poder distinguirla. Su sintaxis es:
- <?xml version="1.0" encoding="UTF-8"?>
  - ► En este ejemplo se indica que se trata de un documento XML
  - Que la versión de XML es la 1.0
  - Y que la codificación de caracteres utilizada es la UTF-8. Para saber más sobre la codificación de caracteres: <a href="https://www.mclibre.org/consultar/htmlcss/html/html-utf8.html">https://www.mclibre.org/consultar/htmlcss/html/html-utf8.html</a>
  - Puede llevar más información pero que se estudiará en temas posteriores.

 El cuerpo es obligatorio y debe contener 1 solo elemento que llamaremos nodo o elemento raíz (root).

Este elemento puede contener otros elemento formando una estructura en árbol:



Así que podemos hablar de nodos padres, hijos, hermanos...

 Un elemento siempre tiene una etiqueta de apertura, un contenido y una etiqueta de cierre:

```
<etiqueta>contenido</etiqueta>
```

En el caso de que el contenido sea prescindible o esté vacío se permiten las siguientes notaciones:

```
<apellido2></apellido2> es equivalente a <apellido2 />
```

 Las etiquetas siempre deben abrirse y cerrarse en el orden correcto. Esto provocaría un error sintáctico:

```
<alumno>Aitor Tilla<edad>25</alumno></edad>
```

- El nombre de la etiqueta debe cumplir las siguientes reglas:
  - Los nombres de los elementos distinguen entre mayúsculas y minúsculas. Esto daría un error sintáctico: <Animal>Gato</animal>
  - Los nombres de los elementos deben comenzar con una letra o un guión bajo
  - Los nombres de los elementos no pueden comenzar con las letras xml (o XML, o Xml, etc.)
  - Los nombres de los elementos pueden contener letras, letras con tilde, dígitos, guiones, guiones bajos y puntos.
  - ► Los nombres de los elementos no pueden contener espacios

Busca los nombres de etiquetas inválidos: <jOsE>, <Álvaro>, <\_Lucía 32>, <.Lucía>, <#Juanito>, <xmlpower>, <\_xmlpower>, <3Barras>, <JohnieWalker>

- Buenas prácticas con los nombres de etiquetas:
- □ Usa nombres descriptivos: <persona>, <empresa>, , , nombres como: <\_32>, <a>, <mietiqueta>, <cosa>...
- Usa nombres cortos y simples: libro\_titulo> y no: <el\_titulo\_del\_libro>.
- □ Evita el uso de "-", "." y ":" en la etiqueta. Son caracteres válidos pero hay lenguajes de programación que usan esos símbolos como operadores (resta, acceso a propiedades...) y pueden provocar errores en el procesado del XML
- Las letras que no estén en el alfabeto inglés (áéíóúü...) son válidas pero puede que el software que procese el XML no esté preparado para soportarlas.
- Utiliza un criterio de nombrado consistente en todo el documento. Por ejemplo:
  - □ Todo en mayúscula/minúscula y el "\_" como separador: <libro\_titulo>
  - □ Todo en minúscula excepto el comienzo de cada palabra: <LibroTitulo>

 Un contenido puede ser: un texto o un número o también uno o más elementos:

```
<alumno>Aitor Tilla</alumno>
<edad>25</edad>
<asignaturas>
    <asignatura>Programación</asignatura>
    <asignatura>Entornos de programación</asignatura>
    <asignatura>LLMM y SGI</asignatura>
</asignaturas>
```

 Para que un texto sea válido como contenido de un documento XML no puede utilizar los símbolos <> porque están reservados para las etiquetas y podrían hacer que el analizador falle... Esto daría un error sintáctico:

<MenorEdad>edad < 18</MenorEdad>

¿Cómo lo se soluciona entonces? Pues XML define unos elementos especiales llamados entidades de caracteres y que representan un carácter concreto pero sin escribirlo directamente.

El caso anterior se arreglaría así:

<MenorEdad>edad &lt; 18</MenorEdad>

- Esta solución también la siguen otros lenguajes como HTML y existe un conjunto grande de entidades de caracteres.
- Algunos ejemplos:

Sintaxis: &NombreEntidad;

Nombre de la Entidad	Carácter representado	Proviene de
lt	<	less than (menor que)
gt	>	greater than (mayor que)
quot	II	quote (cita)
amp	&	ampersand (idem)
apos	'	apostrophe (apóstrofe)

Cuando el analizador del documento XML se encuentra con una entidad de carácter sabe que tiene que cambiarla por el carácter que representa.

## 8 - XML

**ANALIZADOR** 

- Si tenemos un texto muy largo esto de poner las entidades de caracteres puede ser una pesadilla. Para estos casos se usan las secciones CDATA (Character Data).
- Una sección CDATA es una porción del documento en la que el analizador no va a buscar etiquetas, es decir, es una sección que el analizador la va mostrar tal y como está escrita.
- ► Tiene una sintaxis un poco rara:

### <![CDATA[

Aquí puedo poner lo que quiera <etiquetas> y todo. Se va a mostrar tal cual está aquí



## 8 - XML

¿Cómo crees que se muestra en un navegador el siguiente documento XML?

REALIZAR LOS EJERCICIOS DEL 2 AL 5 DEL BOLETÍN DE PROBLEMAS

• Dijimos anteriormente que las etiquetas de apertura opcionalmente pueden llevar uno o más atributos.

Siguen la siguiente sintaxis:

libro id="23442">El Señor de los Anillos

<alumno dni="28748415T" altura="178">Juan Sin Miedo</alumno>

Los atributos son pares **nombre="valor"** que sirven para aportar una información de detalle extra a la etiqueta que lo contiene. Es lo más parecido a un post-it sobre un documento.

Los atributos solo pueden contener **un valor**. No se permite lo siguiente:

<alumno telefonos="651234234, 954342323">Juan Sin Miedo</alumno>





### Compara:

```
<person gender="female">
   <firstname>Anna</firstname>
   <lastname>Smith</lastname>
   </person>
```

```
<person>
    <gender>female</gender>
    <firstname>Anna</firstname>
    <lastname>Smith</lastname>
</person>
```

¿ Qué opción pensáis que es mejor para organizar la información?

- No hay reglas de cuándo utilizar atributos o cuándo utilizar elementos. Sin embargo, podemos decir que:
  - Los atributos son más fáciles de leer para los humanos que los elementos.
  - Los atributos están más limitados que los elementos porque no admiten la estructura en árbol o tener más de un valor.
  - Los elementos se expanden más fácilmente que los atributos.



## 8 - XML

- Sin embargo, en HTML se usan muchos atributos por dos razones:
- Son más legibles y más compactos (ocupan menos)
- Se utilizan para etiquetar a las etiquetas, es decir se usan como metadatos (datos sobre los datos).

De este modo, un programa que maneje el documento XML puede editar o eliminar elementos identificándolos por su atributo.

```
<messages>
  <note id="501">
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this weekend!</body>
  </note>
  <note id="502">
    <to>Jani</to>
    <from>Tove</from>
    <heading>Re: Reminder</heading>
    <body>I will not</body>
  </note>
</messages>
```

Borrar la nota con id="502"

**PROGRAMA** 

 Los comentarios son anotaciones que hace la persona que escribe el documento XML. Su sintaxis es la siguiente:

```
<!--Esto un comentario en XML-->
```

Son elementos ignorados por el analizador (parser) y siguen las siguientes reglas:

- No pueden aparecer antes del prólogo.
- No se pueden anidar un comentario dentro de otro.
- Pueden aparecer en cualquier parte de un documento excepto dentro de los valores de los atributos.

REALIZAR LOS EJERCICIOS 6-8 DEL BOLETÍN DE PROBLEMAS

- ▶ 8.4 Espacios de nombres en XML
- Supongamos que tenemos una empresa internacional (<u>www.interinvest.com</u>) que ofrece servicios de inversión en bolsa en las capitales de todos los países del mundo.
- Una inversión en bolsa viene especificada por los siguientes datos:

Además, nuestra empresa toma información del INE (Instituto Nacional de Estadística) para conocer las distintas capitales de los países, recibiendo documentos como este:

¿Cómo podríamos ampliar el elemento inversión para añadir la capital en la que se invierte? ¿Qué ocurriría con las etiquetas capital? ¿Cómo sabemos si nos referimos al dinero o a la ciudad?



## 8 - XML

Es ambiguo. Las máquinas y las cosas ambiguas se llevan mal...

- Necesitamos algo que rompa la ambigüedad...
- ► ¡Un prefijo nos iría bien!. Por ejemplo:
  - i:capital se refiere al dinero (i=inversión)
  - p:capital se refiere a la ciudad (p=país)
- … lo escribimos y cruzamos los dedos a ver si funciona...

- Nos pide una cosa que se llama "espacio de nombres"
- Este concepto se usa muchísimo en programación y en administración de sistemas y cuanto antes lo tengáis claro mejor...



- Namespace o espacio de nombres: es un contenedor de nombres que cumple las siguientes reglas:
- El namespace tiene un identificador único en el mundo. No puede haber 2 que se llamen igual.
- Un namespace pude contener muchos nombres pero estos no pueden repetirse.

### **NS 1**

- Inversión
- Capital
- Persona

### NS 2

- País
- Capital
- Habitante

### **NS 3**

- Hogar
- Persona
- País

- De esta forma no hay ambigüedades porque:
- NS1:Capital ≠ NS2:Capital
- NS1:Persona ≠ NS3.Persona
- NS2:País ≠ NS3:País

### NS 1

- Inversión
- Capital
- Persona

### NS 2

- País
- Capital
- Habitante

#### NS 3

- Hogar
- Persona
- País

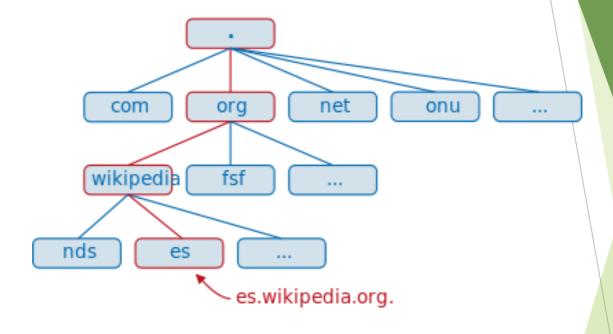


¿Pero quién se encarga de organizar esos NS1, NS2, NS3... para que no se repitan? ¿Quién se responsabiliza de que todos los namespaces del mundo tengan un identificador único?

¿Y si alguien ya lo está haciendo y podemos aprovechar su trabajo?

## 8 - XML

► ICANN (Internet Corporation for Assigned Names and Numbers) es una organización que controla que los dominios de Internet no se repitan (entre otras cosas).





Simplemente aprovechando que los nombres de dominio son únicos en el mundo tendríamos nuestro problema resuelto. Veamos cómo queda:

Nuestro documento XML entonces quedaría así y sería correcto sintácticamente:

- Los analizadores XML no buscan información ni nada en el nombre de dominio que se indica.
- Sin embargo, un documento XML serio apuntaría a un fichero en el que se describa la etiqueta en cuestión.

Lo más importante es que te quedes con el concepto claro porque te lo vas a encontrar a menudo.

¿Sabrías explicar qué es un espacio de nombres en XML?

