



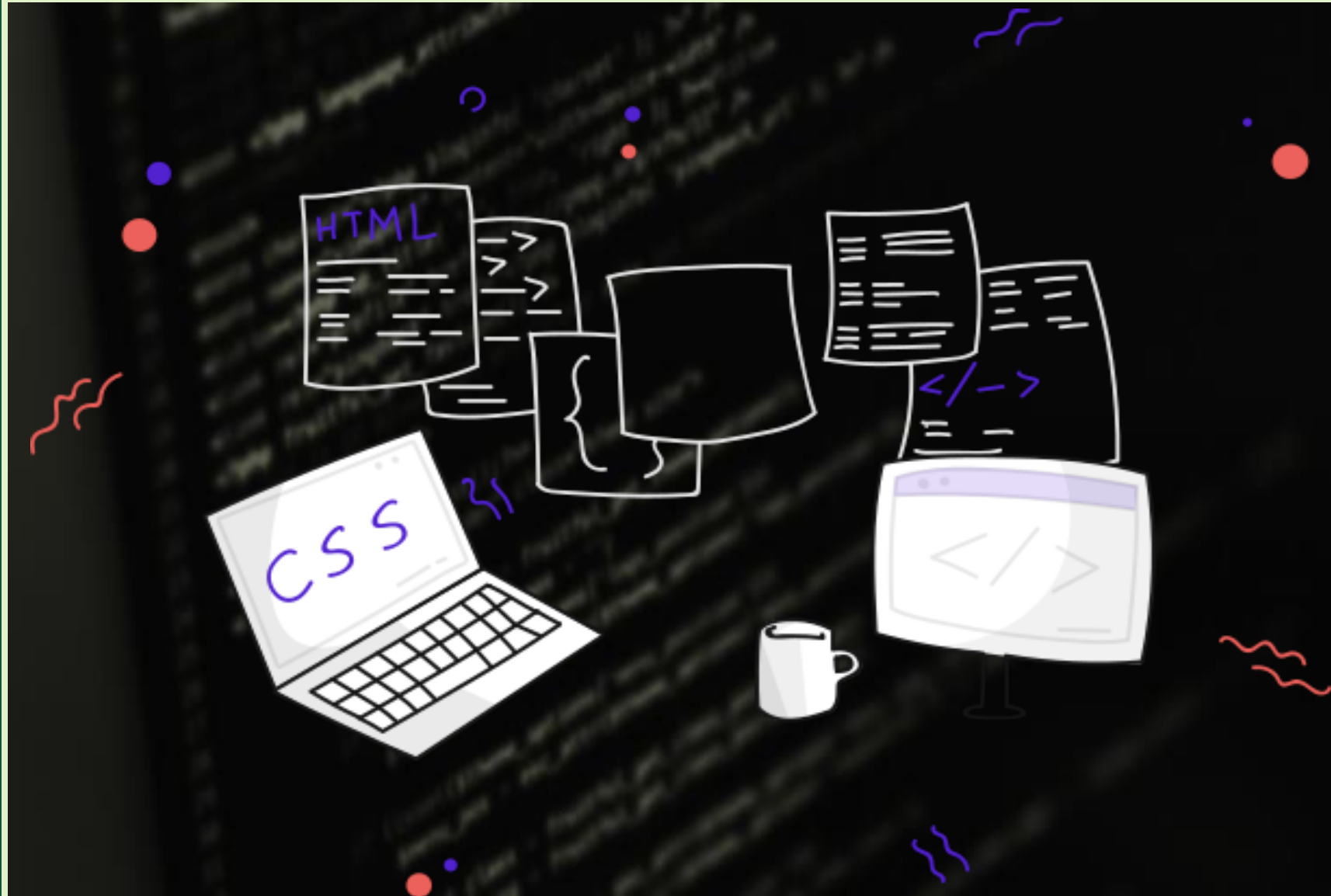
EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Web*

Lenguajes de Marcas y Sistemas de Gestión de la información

María Fátima Sánchez Fuentes

UT1 – INTRODUCCIÓN A LOS LENGUAJES DE MARCAS. XML.





EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Web*

Lenguajes de Marcas y Sistemas de Gestión de la información

INDICE

UT1 – INTRODUCCIÓN A LOS LENGUAJES DE MARCAS. XML.

1. ¿Qué son los lenguajes de marcas?
2. Orígenes y evolución de los lenguajes de marcas.
3. Clasificación de lenguajes de marcas.
4. ¿Qué es XML? Objetivos.
 - 4.1 Separación semántica – representación.
 - 4.2 XML-data y Sintaxis del marcado.
 - 4.3 Etiquetas. Reglas de formación de etiquetas.
 - 4.4 Estructura de un documento XML.
 - 4.5 Elementos. Atributos de los elementos.
 - 4.6 XML *Namespaces*.
 - 4.7 XML bien formado.

¿Qué son los lenguajes de marcas?

1

Lenguajes de Marcas = Lenguajes de etiquetaje = Lenguajes de marcado

Son lenguajes que **combinan**:

- **Información** (generalmente de carácter textual)
- **Marcas** relativas a la **estructura** y **forma** de representar el texto.

Marcas = Etiquetas = Tags

Las marcas no son parte del texto → Información **ADICIONAL**.



Los archivos que contienen marcas/etiquetas/tags comparten las siguientes **características**:

Usan **etiquetas** que
permiten **marcar el
texto**

Separan el **contenido**
de la **presentación**

Son archivos de texto
plano, **editables con
cualquier editor de
texto**

Portables
(Independientes de
plataforma, SSOO o
programa)

Facilitan la interoperabilidad

Son **flexibles** (varios lenguajes
en un único archivo)

Facilidad de **procesamiento**
(más ágil que solo texto plano)

¿Qué son los Lenguajes de Marcas

Ejemplo XML

```
<?xml version="1.0"?>
<quiz>
  <qanda seq="1">
    <question>
      Who was the forty-second
      president of the U.S.A.?
    </question>
    <answer>
      William Jefferson Clinton
    </answer>
  </qanda>
  <!-- Note: We need to add
  more questions later.-->
</quiz>
```

XML

```
▼ <persona profesion="cantante">
  ➡ <nombre>Elsa</nombre> ⬅
  <mujer/>
  ▼ <fecha_de_nacimiento>
    <dia>18</dia>
    <mes>6</mes>
    <año>1996</año>
  </fecha_de_nacimiento>
  <ciudad>Pamplona</ciudad>
</persona>
```

Ejemplo HTML

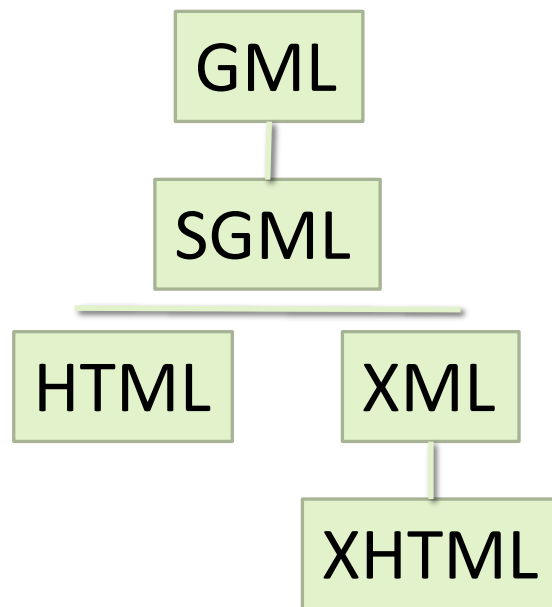
```
<HTML>
<html>
<title>HTML</title>
<body>
  This is HTML!
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="es-ES">
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de 2 párrafos</title>
  </head>
  ➡ <body>
    <p>Esto es un párrafo.</p>
    <p>Esto es otro párrafo.</p>
  </body> ⬅
</html>
```

Orígenes y Evolución de los Lenguajes de Marcas

2

Orígenes y Evolución de los Lenguajes de Marcas



Documento Físico



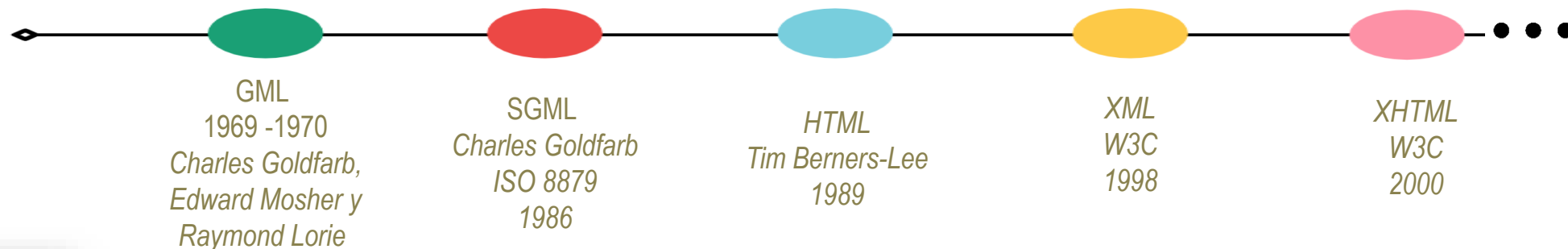
Documento Digital

Metainformación

Al principio los metadatos eran macros o “código específico”

¡NECESIDAD de “código genérico” para interoperabilidad de los documentos!

Lenguajes de Marcas



Clasificación de los Lenguajes de Marcas



De PRESENTACIÓN	De PROCEDIMIENTO	DESCRIPTIVO/SEMÁNTICO
<ul style="list-style-type: none"> Indica el formato del texto. Cómo ha de presentarse el documento. Las etiquetas de marcado suelen estar ocultas al usuario. 	<ul style="list-style-type: none"> Enfocado hacia la presentación del texto. Las etiquetas de marcado son visibles para el usuario. El programa interpreta el código en el mismo orden en el que aparecen. 	<ul style="list-style-type: none"> Usan etiquetas para describir los fragmentos del texto. No especifican como debe presentarse o en qué orden.
<p><u>PEJ:</u> Indicando que una palabra debe presentarse en fuente negrita.</p> <p><u>Ejemplo herramienta:</u> Microsoft Word.</p>	<p><u>PE:</u> Se quiere formatear un título.</p> <div data-bbox="922 981 1663 1153"> <div>Directivas formato título</div> <div>TÍTULO</div> <div>Directivas finalización efectos formato título</div> </div> <p><u>Ejemplo herramienta:</u> HTML, LaTeX.</p>	<p><u>Ejemplo herramienta:</u> XML.</p>

¿Qué es XML?

Objetivos.



XML es un “METALENGUAJE”



XML siempre ha destacado en 3 ámbitos:

- Intercambio de datos.
- Almacenamiento de información.
- Web semántica.



En 1996, el Consorcio Web (W3C) empieza a trabajar para conseguir un lenguaje que combinara:

- **Flexibilidad** de SGML.
- **Aceptación** de HTML.

En Febrero de 1998 se publica XML versión 1.0.



XML (Lenguaje de Marcado Extensible)

XML está diseñado para:

- Almacenar datos.
- Transportar datos.
- Ser legible (por humanos y máquinas).

XML es:

- Un **subconjunto de SGML** simplificado.
- Un **metalenguaje** que permite definir lenguajes de marcado.
- Un conjunto de tecnologías orientadas a **crear nuestros propios lenguajes** de marcas (“vocabularios”).

Vocabularios XML: Descripciones de datos XML usados en el intercambio de información, dentro de un dominio concreto (PEJ matemáticas, química...).

Separación semántica - representación

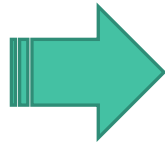




```
<h1 align="center">\n<font family="cursive" color="#061>\n<b><i>EFA MORATALAZ</i></b>\n</font>\n</h1>
```

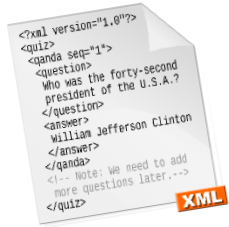
¿Qué es exactamente “EFA MORATALAZ”?

- **Título:** alineado al centro.
- **Fuente:** es cursiva.
- **Color:** verde (#061).
- **Contenido:** “EFA MORATALAZ”.

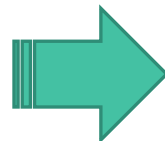


- **PÉRDIDA SEMÁNTICA** del documento.
- **GANANCIA INFORMACIÓN** en la presentación del texto.

XML permite **INDEPENDIZAR** el contenido (información) de un documento XML de su representación (visualización); es decir, en un documento XML...



Encontraremos...	No encontraremos...
Descripción de los datos que conformar el documento XML	Referencias a la presentación de los datos que conforman el documento XML



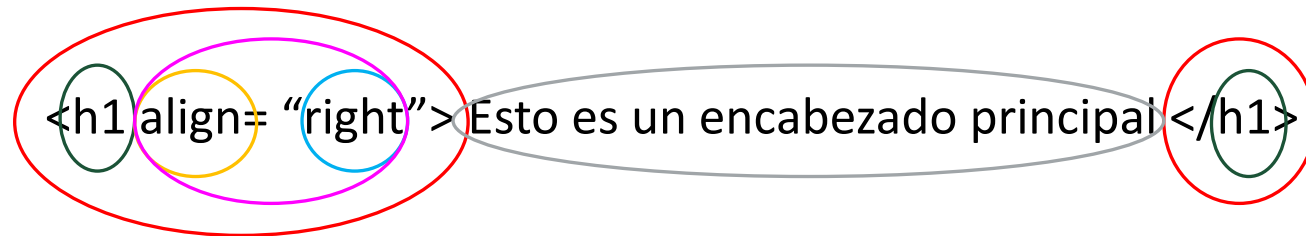
- **GANANCIA SEMÁNTICA** del documento.
- **PÉRDIDA INFORMACIÓN** en la presentación del texto.

Etiquetas. Reglas de formación de etiquetas.



La sintaxis para el marcado XML es muy similar a las de HTML (ya que ambas provienen de SGML).

Elementos de una ETIQUETA:



Nombre del Tag

Tag de apertura/cierre

Atributo del Tag

{ Nombre del atributo
Valor del atributo

Contenido del Tag



Reglas de formación de los identificaciones de los Tags:

1. Los identificadores del Tag son **Key Sensitive**.
2. Siempre empiezan a escribirse **después** del símbolo de apertura: "<".
3. Solamente pueden contener **letras** o **números**, y los siguientes símbolos: "_", ":", "-", ".", ". Siendo siempre el primer carácter una letra o "_".
4. Los Tags deben tener el **mismo nombre** en la apertura y en el cierre.
5. Los Tags se abren y cierran **en orden** (primero se cierra el último que se abrió).
6. Existen los Tags **no duales** (también llamado vacíos o simples).

1. Los identificadores del Tag son **Key Sensitive**.

<LDM> <ldm> <LdM> <lDM> <LDm>

¡Los identificadores de etiquetas son sensibles a mayúsculas y minúsculas!

<LDM> != <ldm> != <LdM> != <lDM> != <LDm>

Es decir, **son identificadores distintos**, <LDM> denota a una identificador distinto que <ldm>.

Nota: Esto no ocurre en HTML.

2. Siempre empiezan a escribirse **después** del símbolo de apertura: "<".



<LDM>

Los identificadores de las etiquetas siempre se comienzan a escribir justo a continuación del símbolo de apertura de etiqueta "<".

3. Solamente pueden contener **letras** o **números**, y los siguientes símbolos:

“_”, “:”, “-”, “.”. Siendo siempre el primer carácter una letra o “_”.

```
<2colores>Rosa y Naranja</2colores>  
<color favorito>azul</color favorito>
```



```
<colores2>Rosa y Naranja</colores2>  
<color_favorito>azul</color_favorito>  
<color.favorito>azul</color.favorito>  
<color-favorito>azul</color-favorito>
```



Los “:” pueden usarse... pero es una palabra reservada.

El “-” y el “.” tampoco es recomendable usarlos debido a que puede causar confusión al combinar XML con otras tecnologías.

4. Los Tags deben tener el **mismo nombre** en la apertura y en el cierre

```
<Ciudad>Ciudad Real</ciudad>  
<ciudad>Ciudad Real</cerrarciudad>  
  
<día>03</dia>  
  
<año>03</anio>
```



Las letras no inglesas (á, Á, ñ, Ñ...) están permitidas. No obstante, no es recomendable usarlas.

5. Los Tags se abren y cierran **en orden** (primero se cierra el último que se abrió).

```
<FONT> <B> <I> 123456 </FONT></B></I>
```



```
<FONT> <B> <I> 123456 </I> </B></FONT>
```



Nota: Esto no ocurre en HTML.

6. Existen los Tags **no duales** (también llamado vacíos o simples).

```
<alumno></alumno>
```

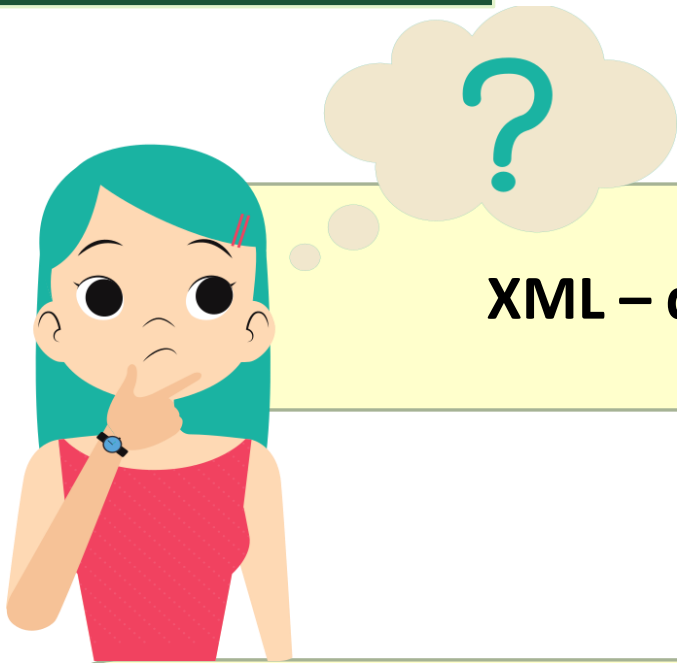
```
<alumno />
```

Ambas opciones son equivalentes.

NOTA: Por regla general, suele ser más común encontrar documentos XML con Tags no duales escritos como en la primera opción.

XML-data y Sintaxis del mercado





XML – data: contenido de un documento XML o paquete de datos.

Sintaxis de mercado: Etiquetas que delimitan la secciones del contenido, las referencias a símbolos especiales y comentarios.

Es decir, nos indica como deben escribirse las etiquetas/tags/marcas, comentarios, símbolos especiales y de presentar la estructura general de los documentos.

XML-data es un contenedor que contiene cadenas de caracteres, entidades especiales, secciones CDATA u otros elementos.

Entidades especiales: caracteres especiales para poner símbolos reservados en el contenido de un Tag.

- Signo "<" se usará <
- Signo ">" se usará >
- Signo "&" se usará &
- Signo "'" se usará '
- Signo "\"" se usará "

CDATA se utiliza para poner un bloque de texto que contiene caracteres y que no deben ser leídos por el *parser* de XML.

<![CDATA [Esto es una cadena de texto que no queremos que el parser interprete. Por eso escribimos > sin tener que ponerlo como carácter especial >]] >

Estructura de un documento XML



PRÓLOGO

Contiene información importante sobre el resto de datos del documento. Es **opcional**.

Está compuesto por:

- **Declaración del XML** → incluye la versión de XML utilizada (actualmente la versión 1.0). Puede incluir el estándar de codificación de caracteres y comentarios o instrucciones de procesamiento.

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

- **Declaración del tipo de documento** → La referencia a la DTD que será usada por el *parser* para comprobar la validez del documento XML, y comentarios o instrucciones de procesamiento.

```
<!DOCTYPE Efa_Moratalaz SYSTEM "http://www.EfaMortalaz.com/validacion.dtd">
```



CUERPO

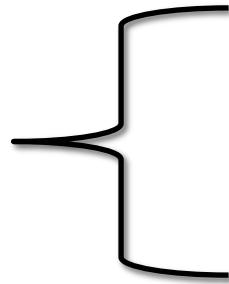
Contiene la **información del documento** (los datos a los que se les ha añadido el marcado).

El cuerpo siempre tiene una estructura de árbol, donde hay un **elemento principal/raíz/"padre"**, el cuál contiene el resto de elementos. Del elemento principal derivan las **ramas del árbol** (elementos "hijos") hasta el nivel más bajo.


Todos elementos de un documento XML pueden tener sub-elementos o elementos "hijos".

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<email>
<para>mfsanchez@efamoratalaz.com</para>
<asunto>Recordatorio</asunto>
<contenido>Buenos días a tod@s, esto es un recordatorio</contenido>
</email>
```


EPÍLOGO



Formado por **comentarios** o **instrucciones de procesado**.
Es **opcional**, puede no tenerlas.



Los identificadores para las etiquetas tienen que ser **significativos y representativos**.

Versión XML que se utiliza (1.0) y el tipo de codificación utilizada ISO-8859-1.

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

```
<email> Marca de apertura del elemento raíz del documento: "email".
```

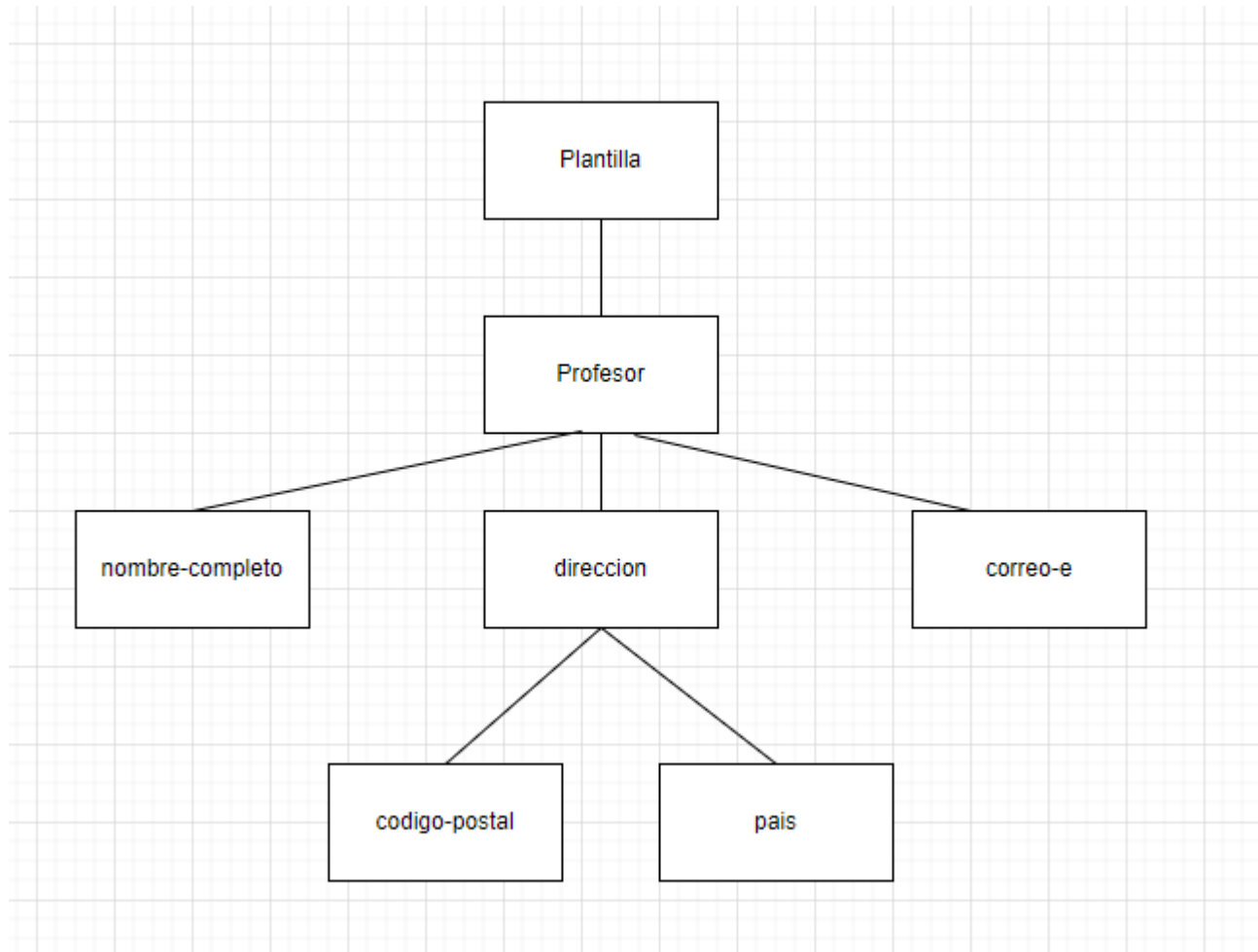
```
<para>mfsanchez@efamoratalaz.com</para> Elemento hijo
```

```
<asunto>Recordatorio</asunto> Elemento hijo
```

```
<contenido>Buenos días a tod@s, esto es un recordatorio</contenido>
```

```
</email> La última línea contiene el cierre del elemento raíz, y el final del documento.
```

EJEMPLO DE ÁRBOL JERÁRQUICO Y XML CORRESPONDIENTE



```
<?xml versión="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE agenda SYSTEM "agenda.dtd">
<plantilla>
  <profesor>
    <nombre-completo>Laura Nieto Torres
    </nombre-completo>
    <direccion>
      <codigo-postal>12541</codigo-postal>
      <pais>España</pais>
    </direccion>
    <correo-e href="abc@gmail.es"></correo-e>
  </profesor>

  <profesor>
    <nombre-completo>Irene Jiménez Santos
    </nombre-completo>
    <direccion>
      <codigo-postal>12541</codigo-postal>
      <pais>España</pais>
    </direccion>
    <correo-e href="def@gmail.es"></correo-e>
  </profesor>
</plantilla>
```

Elementos. Atributos de los elementos



Elemento: Bloques de construcción de un XML.

El contenido de un elemento es todo aquello que se encuentra entre las etiquetas de apertura y cierre. Es decir, los elementos se comportan como una especie de **contenedores** que pueden albergar texto, atributos, o incluso elementos (los cuales reciben el nombre de elementos hijos).

¡OJO! Pueden existir los **elementos vacíos**, que son aquellos que no tienen contenido.

```
<pedido>  
  <pagado/> ← Elemento vacío  
  <contenido>Pedido de cosas</contenido>  
</pedido>
```

Elemento no vacío →

Atributos: Proporcionan metainformación sobre un elemento en un documento XML concreto.

Los atributos proporcionan información **adicional** sobre un elemento.

En otros términos... Son como los *adjetivos* de un elemento.

```
<libros>
<libro>
<importante pagina="1"/>
<importante pagina="5"/>
<importante pagina="120"/>
<importante pagina="201"/>
<importante pagina="225"/>
</libro>
</libros>
```

```
<libros>
<libro titulo="Aprendiendo XML" precio="8">
<importante pagina="1"/>
<importante pagina="5"/>
<importante pagina="120"/>
<importante pagina="201"/>
<importante pagina="225"/>
</libro>
</libros>
```

Entonces... ¿Qué es un elemento y qué es un atributo?

Elemento

```
<libros>
<libro>
<importante pagina="1"/>
<importante pagina="5"/>
<importante pagina="120"/>
<importante pagina="201"/>
<importante pagina="225"/>
</libro>
</libros>
```

Atributo





Reglas de formación de atributos en XML:

1. Los **identificadores** de los atributos se forman de la **misma manera** que los **identificadores** de los **elementos**.
2. El **valor** de los atributos siempre va **entrecomillado** (simples o dobles).
3. **No** se puede repetir el **nombre de un atributo** dentro de un **mismo elemento**.
4. Siempre se especifica el nombre en el Tag de **apertura**.

XML Namespaces



Namespaces: También conocido como *espacio de nombres*. Este agrupa un conjunto de nombres que comparten un mismo contexto.

SINTAXIS:

xmlns:prefix="namespace"

- *namespace* es un URI único.
- *prefix* representa el espacio de nombres y proporciona un puntero al mismo.

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

Podemos usar los espacios de nombres añadiendo el **prefijo** a un determinado (cualquiera) **elemento** del documento XML.

Si hay más de un namespace que vamos a usar en un mismo documento XML...

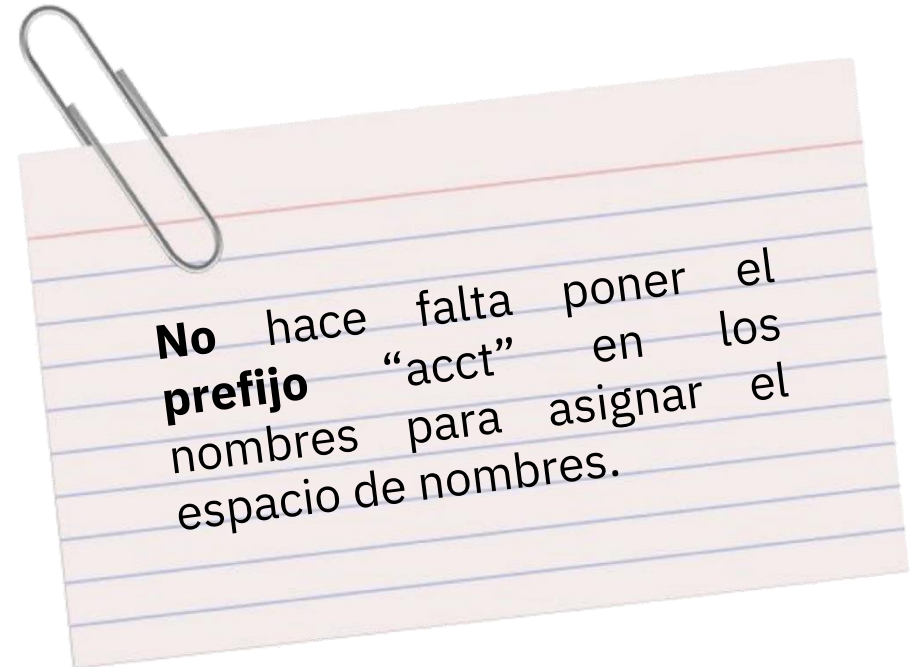
```
<acct:customer xmlns:acct="http://www.my.com/acct-REV10">  
  <acct:name>Corporation</acct:name>  
  <acct:status>invoice</acct:status>  
</acct:customer>
```

Definición del espacio de nombres
Se asigna el espacio de nombres
"http://www.my.com/acct-REV10" al
prefijo "acct".

El prefijo "acct" se utiliza en los
nombres de elementos para
poder asignarles el espacio de
nombres.

Si solo hay un namespace que vamos a usar en un documento...

```
<acct:customer xmlns="http://www.my.com/acct-REV10">  
<name>Corporation</name>  
<status>invoice</status>  
</customer>
```



XML bien formado



XML bien formado: se dice cuando el documento XML cumple con la especificación XML.

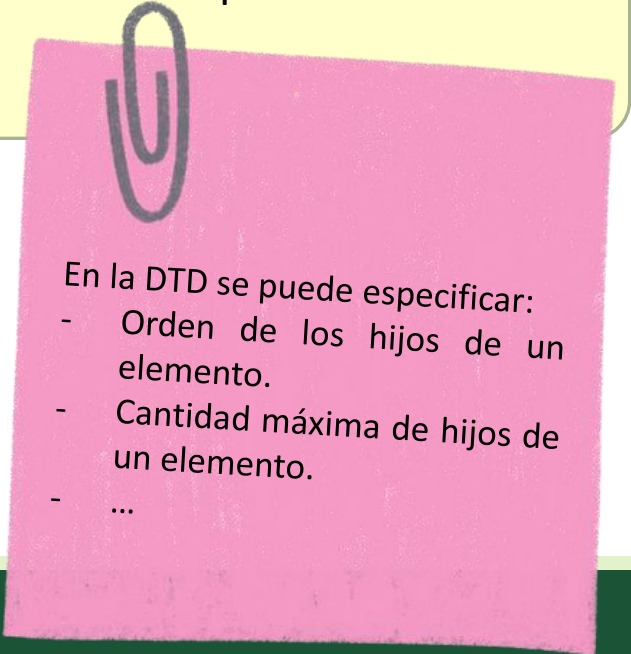
Reglas que deben cumplir los documentos bien formados:

1. El documento debe tener **un solo elemento raíz**.
2. Todas las **etiquetas** (tags) **abiertas** deben tener su respectivas etiquetas de **cierre**.
3. XML distingue **mayúsculas de minúsculas** por lo que todos los elementos y atributos deben seguir la definición.
4. Todos los elementos deben estar **correctamente anidados**.
5. Los **valores de los atributos** deben ir entre **comillas** simples o dobles.
6. Cualquier **elemento Vacío** debe terminar con **/>** o debe hacerlo no vacío añadiendo una etiqueta de cierre.
7. **No se pueden repetir atributos en un mismo elemento.**

En general, un documento XML puede estar en uno de estos estados:

1. **Mal formado** → El peor de los casos.
2. **Bien formado.**
3. **Válido** → está bien formado y además nos han dado las reglas para determinar si algo está bien o mal y el documento XML cumple dichas reglas (DTD).

DTD (Declaración de Tipo de Documento): Es un mecanismo para expresar las reglas sobre lo que se va a permitir y lo que no en archivos XML.

- 
- En la DTD se puede especificar:
- Orden de los hijos de un elemento.
 - Cantidad máxima de hijos de un elemento.
 - ...