

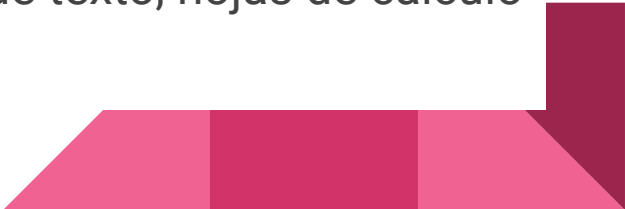
# Tema 04 - XML

## Parte 1

# 1. Introducción a XML

XML (eXtensible Markup Language, Lenguaje de Marcado eXtensible) es un lenguaje utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

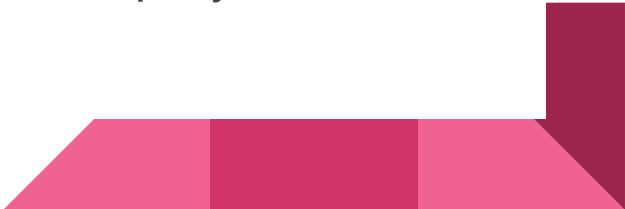


# 1. Introducción a XML

```
<?xml version="1.0" encoding="UTF-8"?>
<clientes>
  <cliente>
    <nombre>AcerSA</nombre>
    <cif>5664332</cif>
  </cliente>
  <cliente>
    <nombre>Mer SL</nombre>
    <cif>5111444</cif>
  </cliente>
</clientes>
```

# 1. Introducción a XML

## Principales usos

- **XML aplicado a los sitios web:** Permite separar el contenido de la presentación y que los mismos datos se puedan mostrar de varias formas distintas sin demasiado esfuerzo.
  - **XML para la comunicación entre aplicaciones:** Representación de los datos muy simple y fácil de transmitir por la red.
  - **Aplicaciones heredadas:** Acceso a datos de aplicaciones que ya no están en uso o incompatibles por diferencias entre versiones.
- 

## 2. Sintaxis de XML

Un documento XML tiene siempre esta estructura básica:

```
Declaración / Versión XML  
Nodo raíz  
  Sub-elementos  
  ...  
Cierre del nodo raíz
```

## 2. Sintaxis de XML

### Declaración XML

Define la versión XML y las características del documento.

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

Esta declaración se escribe al principio de un documento XML, empieza con los caracteres “<?” y termina con “?>” al igual que las instrucciones de procesamiento. Tiene que aparecer en la primera línea del documento, y el carácter “<” debe ser el primero de dicha línea, es decir, antes no puede aparecer nada, ni siquiera espacios en blanco.

## 2. Sintaxis de XML

### Declaración XML

Esta etiqueta tiene los siguientes atributos:

- **version:** Obligatorio a no ser que esté ya incluido en otro documento.
- **encoding:** Opcional. Es la codificación de caracteres con la que se ha guardado el documento.
- **standalone:** Opcional. Indica si un documento va acompañado de un DTD o no lo necesita. Puede tomar dos valores ("yes" o "no").

**standalone="yes":** se está indicando que el documento es independiente de otros, como por ejemplo de una DTD externa.

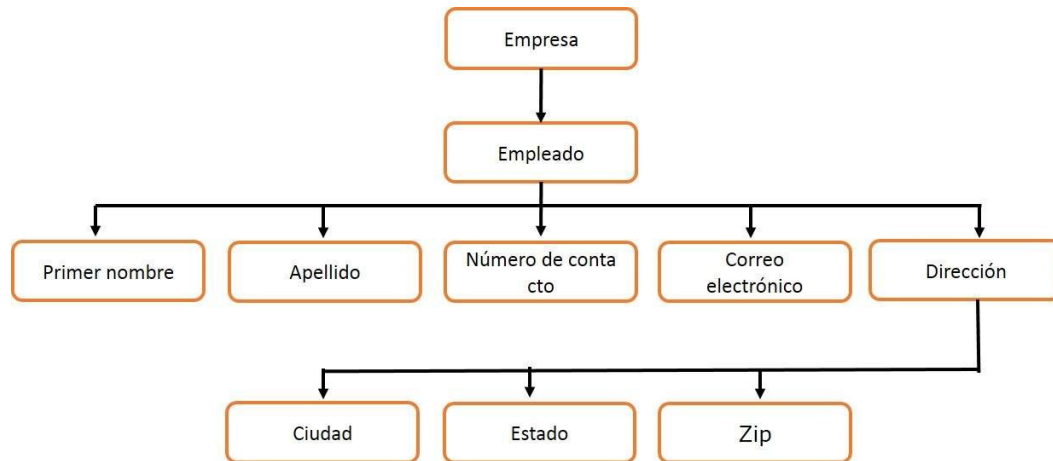
**standalone="no":** significa que el documento no es independiente.

## 2. Sintaxis de XML

### Nodo raíz

Todos los elementos de un documento XML deben estar contenidos entre las etiquetas de apertura y cierre del nodo raíz, es decir, cuelgan del nodo padre que contiene a todos los demás como si fuera una estructura de árbol.

```
<raiz>
  <hijo>
    <nieto> ... </nieto>
  </hijo>
</raiz>
```

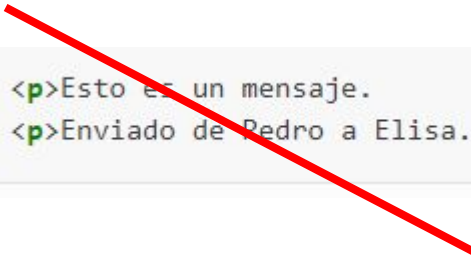




## 2. Sintaxis de XML

### ¡Importante! → Etiquetas de cierre

En XML es sintácticamente ilegal omitir la etiqueta de cierre, a diferencia de en HTML donde algunos elementos pueden no tener etiqueta de cierre.



```
<p>Esto es un mensaje.  
<p>Enviado de Pedro a Elisa.
```

```
<p>Esto es un mensaje.</p>  
<p>Enviado de Pedro a Elisa.</p>
```

Ahora, en el caso de elementos vacíos, se admite una única etiqueta en lugar del par de etiquetas de apertura/cierre. En esos casos, la etiqueta debe escribirse como `<etiqueta />` (poniendo el carácter de etiqueta de cierre después del nombre de la etiqueta).

## 2. Sintaxis de XML

**¡Importante! → Etiquetas de cierre**

**La declaración del documento XML no tiene etiqueta de cierre.** No es un error, simplemente la declaración no forma parte del documento XML y no debe tener etiqueta de cierre.



## 2. Sintaxis de XML

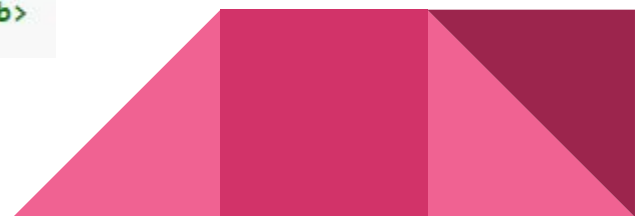
### Anidamiento de etiquetas

El anidamiento incorrecto de etiquetas no tiene sentido en XML.



```
<b><i>Este texto se ve en letra cursiva y negrita</b></i>
```

```
<b><i>Este texto se ve en letra cursiva y negrita</i></b>
```



### 3. Construcción de XML

Para crear un documento XML es importante tener en cuenta una serie de reglas:


- XML es «case-sensitive», es decir que no es lo mismo mayúsculas que minúsculas y que por tanto no es lo mismo <cliente>, que <Cliente> que <CLIENTE>.
- En general, la costumbre es poner todo en minúsculas.
- Obligatorio: solo un elemento raíz.
- Solo se puede poner una etiqueta que empiece por letra o \_. Es decir, esta etiqueta no funcionará en los programas <12Cliente>.
- Aparte de eso, una etiqueta sí puede contener números, por lo que esta etiqueta sí es válida <Cliente12>.
- Aunque no es obligatorio a menudo se suele poner en la primera línea un prólogo que indica la versión de XML que estamos usando y la codificación con la que nuestro editor almacena los archivos.

## 4. Validez

Un documento XML puede «estar bien formado» o «ser válido». Se dice que un documento «está bien formado» cuando respeta las reglas XML básicas. Si alguien ha definido las reglas XML para un vocabulario, podremos además decir si el documento es válido o no, lo cual es mejor que simplemente estar bien formado.

En general, podemos asumir que un documento puede estar en uno de estos estados que de peor a mejor podríamos indicar así:

- Mal formado (lo peor)
- Bien formado.
- **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. Este es el mejor caso.**



Indica por qué los siguientes  
códigos no son correctos

```
<clientes>
  <cliente>
    <nombre>AcerSA
    <CIF>5666333</CIF>
  </cliente>
</clientes>
```

```
<clientes>
  <cliente>
    <nombre>AcerSA</nombre>
    <cif>5666333</CIF>
  </cliente>
</clientes>
```



```
<clientes>
  <cliente>
    <nombre!>AcerSA</nombre!>
    <CIF>5666333</CIF>
  </cliente>
</clientes>
```

```
<cliente>
```

```
  <nombre>AcerSA</nombre>
```

```
  <CIF>5666333</CIF>
```

```
</cliente>
```

```
<cliente>
```

```
  <nombre>ACME</nombre>
```

```
  <CIF>455321</CIF>
```

```
</cliente>
```

## Solución:

1. La etiqueta <nombre> no está cerrada.
2. Se ha puesto <cif> cerrado con </CIF> (mayúsculas).
3. Se ha utilizado la admiración, que no es válida (de hecho, el coloreador de sintaxis automático descubre que no es XML y el fichero se muestra de manera literal).
4. El problema es que hay más de un elemento raíz.

**Construir un documento XML para el siguiente pedido que ha recibido por correo electrónico una empresa que se dedica a la venta de herramientas para jardinería:**

"Hola, necesito una cortadora de césped para mi jardín de esas que anuncian en oferta, me gustaría que fuera uno de esos modelos eléctricos, pues las de gasolina contaminan mucho. Me llamo Roberto Movilla, la cortadora la tendrán que enviar a Albacete, la dirección es Plaza de la Duquesa 12, la recogerá mi esposa que se llama Alicia Abad. Ahora que lo pienso también necesitaré 3 podadoras para los setos. Les paso mi dirección de aquí para lo referente al pago, es General Ricardos 56, aquí en Madrid. Es urgente, por favor, el césped está muy alto."

**Indicar Destinatario, Ordenante, Observaciones, Contenido y en su interior Productos**

```
<?xml version="1.0" encoding="UTF-8"?>
<Pedido>
  <Destinatario>
    <Nombre>Alicia Abad</Nombre>
    <Direccion>Plaza de la Duquesa 12</Direccion>
    <Ciudad>Albacete</Ciudad>
    <CodPostal>05020</CodPostal>
  </Destinatario>
  <Ordenante>
    <Nombre>Roberto Movilla</Nombre>
    <Direccion>General Ricardos 56</Direccion>
    <Ciudad>Madrid</Ciudad>
    <CodPostal>28055</CodPostal>
  </Ordenante>
  <Observaciones>Es urgente, el césped está muy alto</Observaciones>
  <Contenido>
    <Producto codigo="872-AA">
      <Nombre>Cortadora de cesped</Nombre>
      <Cantidad>1</Cantidad>
      <Precio>148.95</Precio>
      <Observaciones>Confirmar que es eléctrica</Observaciones>
    </Producto>
    <Producto codigo="926-FH">
      <Nombre>Podadora</Nombre>
      <Cantidad>3</Cantidad>
      <Precio>7.98</Precio>
    </Producto>
  </Contenido>
</Pedido>
```

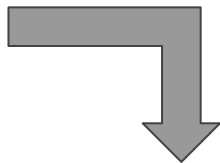
## 5. Atributos

Los elementos de un documento XML pueden tener atributos definidos en la etiqueta de inicio. Un atributo sirve para proporcionar información extra sobre el elemento que lo contiene.

### EJEMPLO

Dados los siguientes datos de un producto:

- Código: G45
- Nombre: Gorro de lana
- Color: negro
- Precio: 12.56



```
<producto codigo="G45">  
  <nombre color="negro" precio="12.56">Gorro de lana</nombre>  
</producto>
```

## 5. Atributos

**¡Importante!** → todos los valores de los atributos deben estar entrecomillados. Pero el tipo de comillas utilizado es irrelevante; podemos utilizar tanto comillas simples como comillas dobles pero, eso sí, debemos utilizar el mismo tipo de comillas en ambas partes de la expresión entrecomillada.

*Incorrecto:*

```
<fichero tipo="gif">mi_casa.gif</fichero>  
<fichero tipo='gif">mi_casa.gif</fichero>
```

*Correcto:*

```
<fichero tipo="gif">mi_casa.gif</fichero>  
<fichero tipo='gif'>mi_casa.gif</fichero>
```

## 5. Atributos


**¡Importante!** → todos los atributos de un elemento tienen que ser únicos.

*Incorrecto:*

```
<datos x="3" x="4" y="5"/>
```

*Correcto:*

```
<datos x="3" X="4" y="5"/>
```





## 5. Atributos

### ¿Elementos o atributos?

```
<persona sexo="femenino">  
  <nombre>Elisa</nombre>  
  <apellido>Lopez</apellido>  
</persona>
```

```
<persona>  
  <sexo>femenino</sexo>  
  <nombre>Elisa</nombre>  
  <apellido>Lopez</apellido>  
</persona>
```

En el primer ejemplo, el sexo es un atributo del elemento persona. En el segundo, sexo es un elemento hijo del elemento persona. No existen reglas sobre cuando utilizar atributos o elementos hijos. Sin embargo, como norma general, se debería tender a utilizar los elementos hijos en lugar de los atributos.

## 5. Atributos

### ¿Elementos o atributos?

El uso de atributos tiene algunos problemas:

- Los atributos son difíciles de expandir en el caso de que se deseen hacer cambios futuros en la estructura de los datos.
- Los atributos no permiten estructurar la información.
- Los atributos son más difíciles de manipular por las aplicaciones.
- Los valores de los atributos son difíciles de verificar frente a una DTD.



## 5. Atributos

### ¿Elementos o atributos?

Sin embargo, hay ocasiones en las que el uso de atributos si puede ser recomendable.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<mensajes>
  <nota ID="001">
    <para>Elisa</para>
    <de>Pedro</de>
    <titulo>Recordatorio</titulo>
    <cuerpo>No olvides nuestra cita!</cuerpo>
  </nota>
  <nota ID="002">
    <para>Juan</para>
    <de>Francisco</de>
    <titulo>Cita</titulo>
    <cuerpo>Quedamos a comer en el Restaurante de abajo.</cuerpo>
  </nota>
</mensajes>
```

*El atributo ID en este ejemplo es solamente un contador de mensajes y no una parte de los datos. En este caso sí podemos decir que el uso de los atributos está recomendado. La información que contiene es los que se denomina metainformación (información sobre la información).*

## 6. Comentarios

Para escribir comentarios en un documento XML, estos deben escribirse entre los caracteres “<!--” y “-->” .

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Ejemplo uso de comentarios.-->
<a>
  <b>
    <c cantidad="4">cccc</c>
    <d cantidad="2">dd</d>
  </b>
  <e>
    <f cantidad="8">ffffffff</f>
    <!--g puede aparecer varias veces.-->
    <g cantidad="5">ggggg</g>
    <g cantidad="2">gg</g>
  </e>
</a>
```

## 7. Caracteres especiales de XML

Hay una serie de caracteres que XML no reconoce y los considera como ilegales. Para poder incluirlos, se utilizan una serie de referencias.

Si por ejemplo introducimos un símbolo de menor < dentro de una etiqueta el parser dará como respuesta un mensaje de error porque considera que si hay un símbolo de menor, es el comienzo de una nueva etiqueta.



```
<mensaje>si salario <1000 entonces </mensaje>
```

## 7. Caracteres especiales de XML

Referencias a entidades en XML		
<i>Carácter</i>	<i>Entidad</i>	<i>Referencia a entidad</i>
< ( <i>menor que</i> )	lt ( <i>less than</i> )	&lt;
> ( <i>mayor que</i> )	gt ( <i>greater than</i> )	&gt;
" ( <i>comilla doble</i> )	quot ( <i>quotation mark</i> )	&quot;
' ( <i>comilla simple</i> )	apos ( <i>apostrophe</i> )	&apos;
& ( <i>ampersand</i> )	amp ( <i>ampersand</i> )	&amp;

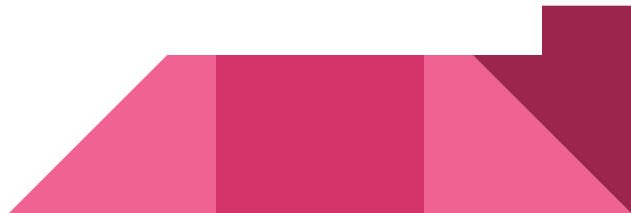
## 7. Caracteres especiales de XML

```
<?xml version="1.0" encoding="UTF-8"?>
<entidades>
  <menor_que>&lt;</menor_que>
  <mayor_que>&gt;</mayor_que>
  <comilla_doble>&quot;</comilla_doble>
  <comilla_simple>&apos;</comilla_simple>
  <ampersand>&amp;</ampersand>
</entidades>
```



## 8. Instrucciones de procesamiento

En un documento XML, una instrucción de procesamiento sirve para indicar cierta información al programa que procese dicho documento. Las instrucciones de proceso se escriben empezando con la pareja de caracteres “<?” y finalizando con “?>”.



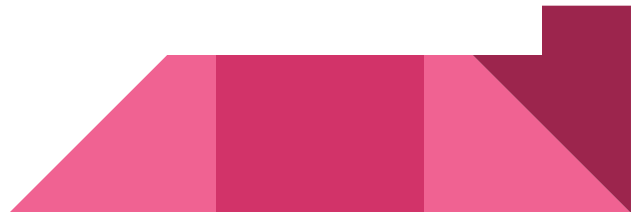


## 8. Instrucciones de procesamiento

### Asociar un archivo CSS a un documento XML

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

Esta instrucción sirve para asociar el archivo CSS “estilo-animales.css” al documento XML.



## 8. Instrucciones de procesamiento

### Asociar un archivo CSS a un documento XML

```
nombre{color:blue;font-size:40px}  
patas{color:red;font-size:22px}
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<?xml-stylesheet type="text/css" href="estilo-animales.css"?>  
<animales>  
  <animal>  
    <nombre>perro</nombre>  
    <patas>4</patas>  
  </animal>  
  <animal>  
    <nombre>pato</nombre>  
    <patas>2</patas>  
  </animal>  
  <animal>  
    <nombre>ballena</nombre>  
    <patas>0</patas>  
  </animal>  
</animales>
```



# Classroom