

## Tema 1. Documentos XML bien formados.

Como ya se comentó en el módulo anterior, para que un documento XML esté correctamente escrito requiere de dos condiciones:

- Ser un documento válido.
- Ser un documento bien formado.

A continuación, se verá qué significan estas dos condiciones.

### DOCUMENTO BIEN FORMADO

Un documento está bien formado si se dan las mínimas condiciones para ser manipulados por las aplicaciones, y para ello se debe cumplir lo siguiente:

- ✍ Debe aparecer una etiqueta XML con:
  - la versión, no es obligatorio pero sí recomendable. Para ello, se utilizará el atributo *versión*.
  - el tipo de codificación, a través del atributo *encoding*, que indicará el formato de los caracteres del documento, por ejemplo: UTF-8
- ✍ Debe haber una etiqueta de inicio y otra de final por cada elemento del XML, salvo cuando sea un elemento vacío (se explicará más adelante) que puede tener en una sola etiqueta inicio y final. Es decir todas las etiquetas deben estar cerradas.
- ✍ Toda la información del documento debe estar agrupada en un elemento principal. Es decir, todas las etiquetas deben estar dentro de una única etiqueta que será el padre del resto.
- ✍ Tenemos que tener en cuenta que el XML es *case sensitive*, es decir distingue las etiquetas mayúsculas y minúsculas, pues para él son distintas.
- ✍ El valor de los atributos debe estar siempre entre comillas simples o dobles.
- ✍ Las etiquetas deben estar bien anidadas, es decir no puede haber elementos solapados. Por ejemplo, si tenemos 2 elementos y uno contiene al otro, no se puede cerrar antes la etiqueta del primero que la del segundo; debe cerrarse la marca del elemento contenido en primer lugar.
- ✍ Las etiquetas no pueden tener un mismo nombre con formatos diferentes.

Ahora, si volvemos a retomar el Ejemplo 2 que se expuso en el anterior módulo, seremos capaces de identificar lo que se acaba de comentar:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro>
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>

  <libro>
    <titulo> EL FUEGO </titulo>
    <autor> KATHERINE NEVILLE </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>

  <libro>
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> STEVE ALTEN </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>

  <libro>
    <titulo> CREPUSCULO </titulo>
    <autor> STEPHENIE MEYER</autor>
    <categoria> Juvenil </categoria>
  </libro>
</biblioteca>
```

Veamos con detalle:

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

Se detalla la versión de XML, en este caso 1.0, y el tipo de codificación que se está utilizando (ISO-8859-1).

Observa como los valores de los atributos se especifican entre comillas.

```
<biblioteca>
....
</biblioteca>
```

Como se ha especificado, existe una etiqueta de apertura y otra de cierre, ya que en este caso no se trata de un elemento vacío.

Observa como el resto de etiquetas, están dentro de <biblioteca> </biblioteca> es decir, toda la información del documento debe estar agrupada en un elemento principal.

```
<libro>
  <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
  <autor> SUSANA VALLEJO </autor>
  <categoria> Juvenil </categoria>
</libro>
```

Observa, cómo se han generado etiquetas para describir los objetos que estamos mostrando, en nuestro caso libros, donde el orden es importante y no se cambian entre mayúsculas y minúsculas.

## DOCUMENTO VÁLIDO

Un documento XML será un documento válido si cumple las restricciones indicadas en su DTD.

Pero, ¿qué es una DTD? Es la Definición de Tipo de Documento, es decir, el conjunto de elementos que se pueden incluir en un documento de un tipo determinado, el orden en el que deben aparecer y cómo deben anidarse se recoge en la llamada Definición de Tipo de Documento o DTD.

Estas restricciones indican qué elementos pueden aparecer en el documento, en qué orden deben hacerlo, cómo pueden anidarse, qué atributos están permitidos para cada elemento, etc.

Para que un documento XML sea válido, es necesario disponer de su DTD para poder hacer las comprobaciones necesarias.

Cada documento XML debe indicar al comienzo el DTD usado por medio de la marca: **<!DOCTYPE nombre del documento SYSTEM "nombre.dtd">**.

En el siguiente módulo se verá con más detalle como se crean las DTDs, de momento pondremos un ejemplo:

Código:

```
<?xml version="1.0">
<!DOCTYPE libro [
  <!ELEMENT libro (titulo, autor, categoria)>
  <!ELEMENT titulo (#PCDATA)>
  <!ELEMENT autor (#PCDATA)>
  <!ELEMENT categoria (#PCDATA)>
]>
<libro>
<titulo>Crepúsculo</titulo >
<autor>STEPHENIE MEYER</autor >
<categoria>Juvenil</categoria >
</libro>
```

En el siguiente módulo se verá con más detalle como generar documentos XML válidos y se explicará con más detalle este ejemplo.

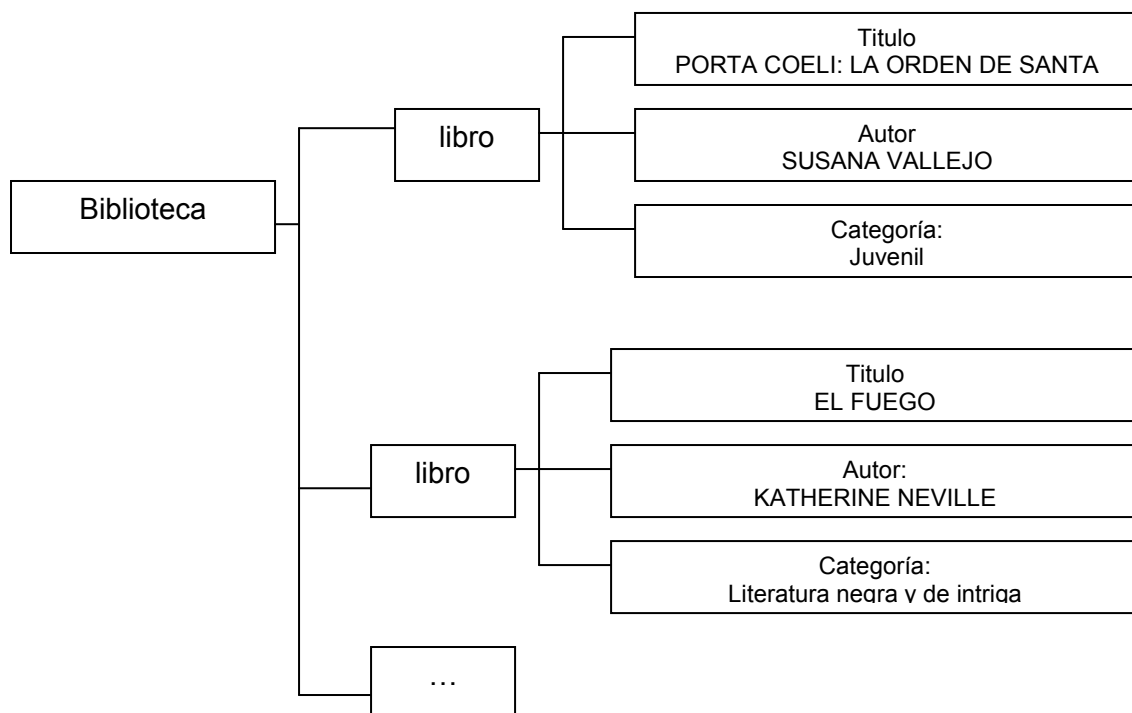


## Tema 2. Partes de un documento XML.

Aunque a primera vista, un documento XML puede parecer similar a HTML, hay una diferencia principal. Un documento XML contiene datos que se autodefinen, exclusivamente. Un documento HTML contiene datos mal definidos, mezclados con elementos de formato. En XML se separa el contenido de la presentación de forma total.

Veamos esto, gráficamente a través del ejemplo de la biblioteca cuyo código XML era:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro>
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro>
    <titulo> EL FUEGO </titulo>
    <autor> KATHERINE NEVILLE </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro>
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> STEVE ALTEN </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro>
    <titulo> CREPUSCULO </titulo>
    <autor> STEPHENIE MEYER</autor>
    <categoria> Juvenil </categoria>
  </libro>
</biblioteca>
```



De esta forma se puede observar más fácilmente como los documentos XML deben tener una estructura jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Esto significa que los elementos deben estar correctamente anidados y que los elementos no se pueden solapar entre ellos. Además, los elementos con contenido deben estar correctamente cerrados.

En conclusión, como resultado de todo lo expuesto hasta ahora, podemos decir que un documento XML se compone de:

➡ Un prólogo:

✍ Declaración XML: los documentos XML deben, aunque no es imperativo, comenzar con una declaración XML.

Ésta se parece a una instrucción de procesamiento con el nombre *xml* y con otros atributos: *version*, *standalone* y *encoding*.

- Atributo *version*: especifica la versión del analizador XML. Para ver las versiones más recientes consultar: <http://www.w3.org/XML>.
- Atributo *standalone*: este atributo puede tomar los valores “yes” o “no” e indicará si tiene asociada un DTD. Por defecto, si no se especifica tendrá el valor “no”.
- Atributo *encoding*, definirá el conjunto de caracteres que vamos a utilizar. Las codificaciones que soporta XML son:
  - UTF-8
  - UTF-16
  - ISO-10646-UCS-2
  - ISO-10646-UCS-4
  - ISO-8859-1 a -9
  - ISO-2022-JP
  - Shift\_JIS
  - EUC-JP

Por ejemplo:

```
<?xml version="1.0" encoding = "UTF-8" standalone = "no" ?>
```

Esta declaración indica que este documento es conforme a la versión 1.0 de la norma XML, precisa el juego de caracteres UTF-8 y no hay de referencias externas.

Los documentos XML no tienen porque tener declaraciones XML. Sin embargo, si la tienen, dicha declaración debe estar al principio del documento, es decir, en la primera línea.

No puede estar precedida de comentarios, espacios en blanco,... ya que el analizador XML utiliza los primeros cinco caracteres `<?xml` para realizar conjeturas razonables para la codificación.

✍ Una posible referencia a un documento DTD por ejemplo :

```
<!DOCTYPE biblioteca SYSTEM "mislibros.dtd">
```

✍ Una posible referencia a un documento Schema XML.

(Estos dos últimos puntos, se explicarán en el siguiente módulo del curso).

➡ Un elemento raíz que contiene los elementos:

```
<biblioteca>
  <libro>
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro>
    <titulo> EL FUEGO </titulo>
    <autor> KATHERINE NEVILLE </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro>
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> STEVE ALTEN </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro>
    <titulo> CREPUSCULO </titulo>
    <autor> STEPHENIE MEYER</autor>
    <categoria> Juvenil </categoria>
  </libro>
</biblioteca>
```

➡ Los comentarios e instrucciones de tratamiento, que se explicarán más adelante.

## Tema 3. Elementos.

Por elemento entendemos el conjunto de fragmentos que componen la etiqueta o *tag* HTML.

Su sintaxis general es:

```
<nombre_etiqueta> Contenido </nombre_etiqueta>
```

Por ejemplo:

```
<Producto> Botella de agua </Producto>
```

En el ejemplo de la biblioteca:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro>
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro>
```

**Elementos**

```
<titulo> EL FUEGO </titulo>
<autor> KATHERINE NEVILLE </autor>
<categoria> Literatura negra y de intriga </categoria>
</libro>
<libro>
  <titulo> LA RESURRECCION MAYA </titulo>
  <autor> STEVE ALTEN </autor>
  <categoria> Literatura negra y de intriga </categoria>
</libro>
<libro>
  <titulo> CREPUSCULO </titulo>
  <autor> STEPHENIE MEYER</autor>
  <categoria> Juvenil </categoria>
</libro>
</biblioteca>
```

Como ya se comentó en el tema anterior, los documentos XML tienen una estructura jerárquica por lo que un elemento puede contener a su vez otros elementos. De esta forma, se conforma esta jerarquía o “árbol” en la que unos elementos contienen a otros.

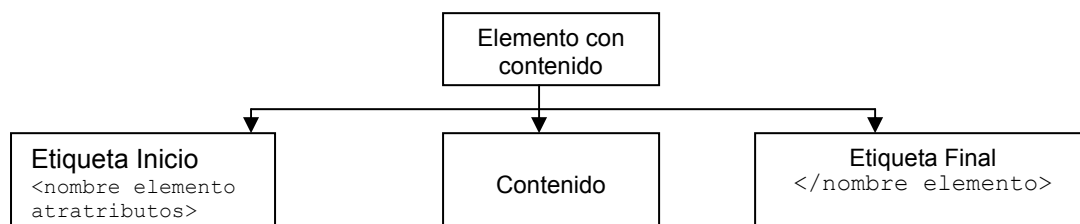
Hay muchos editores de XML, como se explicará en el módulo 4, ofreciendo la mayoría de ellos la posibilidad de ‘navegar’ por esta estructura jerárquica.

Por tanto, habrá un tipo de elemento padre de todos que contendrá al resto de elementos. Se trata del elemento raíz. En nuestro ejemplo, se trata del elemento `<biblioteca></biblioteca>`. Las características de un elemento raíz son:

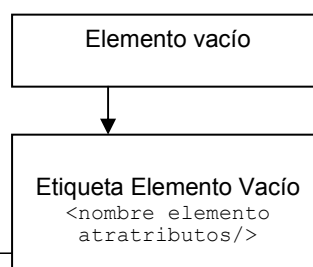
- ✍ Incluye a todos los elementos del documento.
- ✍ No está comprendido en ningún otro elemento.
- ✍ Su nombre debe coincidir con el nombre del tipo de documento.

Podemos distinguir entre:

- elementos con contenido, como en el ejemplo anterior.



- elementos vacíos





## ELEMENTOS VACÍOS

Un documento XML puede contener un tipo especial de elementos: los elementos vacíos.

Estos elementos no tienen ningún contenido. Es decir, entre sus marcas de inicio y de fin no se escribe nada.

Los elementos vacíos se utilizan para posicionar elementos en el documento: por ejemplo para incluir imágenes, archivos multimedia, etc.

Para añadir la información necesaria para que una aplicación pueda gestionar un elemento vacío se añaden atributos al elemento. En el siguiente tema, se explicarán qué son los atributos.

Los elementos vacíos se pueden incluir en un documento utilizando dos sintaxis:

- La primera consiste en utilizar una marca de inicio y otra de fin, sin ningún contenido entre ellas. Por ejemplo:

```
<imagen nombre="portada" formato="gif"></imagen>
```

- La segunda sintaxis utiliza una única marca. Para indicar que se trata de un elemento vacío se añade una barra invertida / antes del carácter de fin de la marca, tal y como se señala en el siguiente ejemplo:

```
<imagen nombre="portada" formato="gif"/>
```

## Tema 4. Atributos

XML permite complementar a un elemento con atributos que matizan su significado.

Cada uno de los elementos que componen nuestro archivo XML, ya posea o no contenido, puede tener atributos que nos ofrecen información sobre el mismo. Su sintaxis general es:

```
<nombre_etiqueta nombre_atributo="valor"
nombre_atributo2="valor">Contenido</nombre_etiqueta>
```

O, para el caso de elementos vacíos:

```
<nombre_etiqueta nombre_atributo="valor" nombre_atributo2="valor"/>
```

Ejemplos:

```
<Producto nombre="Botella de Agua"/>
<Producto nombre="Botella de Agua" precio="1€" unidades="4"/>
<Producto nombre="Botella de Agua">1€</Producto>
```

Es necesario tener en cuenta:

- ✍ Los atributos permitidos para cada elemento se deben indicar en la DTD, si la tiene.
- ✍ Los atributos tienen un nombre y reciben un valor con un tipo de dato especial.
- ✍ Los atributos se escriben dentro de la marca de inicio del elemento, en cualquier orden, tras el nombre del elemento.
- ✍ Tras el nombre del atributo se escribe el signo igual y el valor que se le asigna escrito entre comillas.

Ejemplo:

**Atributo**

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro id="Juv-010">
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro id="Int-011">
    <titulo> EL FUEGO </titulo>
    <autor> KATHERINE NEVILLE </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Int-211">
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> STEVE ALTEN </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Juv-917">
```

```
<titulo> CREPUSCULO </titulo>
<autor>   STEPHENIE MEYER</autor>
<categoria> Juvenil  </categoria>
</libro>

</biblioteca>
```

## VALORES DE LOS ATRIBUTOS

Los valores que pueden tomar los atributos son:

- CHAR: es decir, datos de tipo texto.
- ENTITY: identificador de una entidad declarada en la DTD (como se verá en el siguiente módulo).
- ID: identificador que debe ser único en el documento. Si hay dos atributos de tipo ID con el mismo valor en un mismo documento, el documento no será válido.
- Listas de opciones: un valor de una lista de valores posibles. Siempre se propondrá un valor por defecto para el atributo.
- ENTITIES - los identificadores de dos o más entidades declaradas en la DTD, separados por comas.
- IDREF: toma como valor el valor de un atributo de tipo ID utilizado en el mismo documento.

## NOMBRES XML

Las reglas para definir los nombres de los elementos XML son las mismas que las reglas para definir los nombres de los atributos de XML, ya que en conjunto se conocen a estos nombres como nombres XML.

Los nombres XML:

- ✍ Puede contener básicamente cualquier carácter alfanumérico: A..Z, a..z, 0..9.
- ✍ Puede incluir caracteres que no pertenezcan al alfabeto, como ö, ç,
- ✍ Puede incluir el guión bajo o carácter subrayado, el guión alto y punto.
- ✍ No pueden contener otros signos como: comillas, signos de puntuación, signo dólar, símbolo porcentaje o punto y coma.
- ✍ No pueden contener un espacio en blanco
- ✍ Todos los nombres que comienzan con la cadena XML, ya sea mayúscula o minúscula, se reserva para especificaciones del estándar XML.

## Tema 5. Entidades predefinidas.

En XML existen algunos caracteres reservados que no podemos utilizar para evitar problemas con el marcado, lo que no significa que no tengan que salir en nuestros documentos XML.

Las entidades predefinidas son marcas XML que se utilizan para representar estos caracteres. El XML especifica cinco entidades predefinidas:

- ✍ `&amp;` para el `&`
- ✍ `&lt;` para el `<`
- ✍ `&gt;` para el `>`
- ✍ `&apos;` para el `'`
- ✍ `&quot;` para el `"`

Como podemos observar, se reconocen al ir entre los símbolos "&" y ";".

Por ejemplo, supongamos el caso que el nombre del autor de los libros del Ejemplo Biblioteca debiera aparecer entre comillas. ¿Cómo lo haríamos? Para XML las `"` se utilizan para especificar los valores de los atributos, ¿cómo sabría que debe mostrar las comillas?

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro id="Juv-010">
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> &quot; SUSANA VALLEJO &quot; </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro id="Int-011">
    <titulo> EL FUEGO </titulo>
    <autor> &quot; KATHERINE NEVILLE &quot; </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Int-211">
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> &quot; STEVE ALTEN &quot; </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Juv-917">
    <titulo> CREPUSCULO </titulo>
    <autor> &quot; STEPHENIE MEYER &quot; </autor>
    <categoria> Juvenil </categoria>
  </libro>
</biblioteca>
```

## Tema 6. Secciones CDATA.

Otra solución, para escribir los caracteres especiales que ya tienen un significado propio para el procesador de XML (como el símbolo <, >, “...”.) que no sean las entidades predefinidas es el uso de secciones CDATA. Es otra construcción en XML que permite especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML.

Es decir, dentro de estas secciones irán todos aquellos caracteres que no queramos que interprete el procesador.

```
<![CDATA[
    Aqui puedo poner lo que quiera.
]]>
```

Los CDATA empiezan con los caracteres "<![CDATA[" y termina con "]]>".

La razón de esta construcción llamada CDATA (Character DATA) es que a veces es necesario para los autores de documentos XML, poder leerlo fácilmente sin tener que descifrar los códigos de entidades. Especialmente cuando son muchos.

Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<mensaje>
<![CDATA[
<parrafo>Esto es un curso de XML en el que
    las etiquetas <> son muy importantes
</parrafo>
]]>
<parrafo>Esto es un curso de XML en el que
    las etiquetas &lt; &gt; son muy importantes
</parrafo>
</mensaje>
```

Todo lo que se escriba dentro de la sección CDATA se mostrará tal cual en el documento, sin interpretarlo el procesador

## Tema 7. Comentarios.

Los documentos XML se pueden comentar, ya sea:

- para añadir información sobre el documento: fecha de creación, autor, etc...
- para dejar información entre los coautores

Los comentarios son estrictamente para conseguir que el código fuente de un documento XML sea más legible para los lectores.

Para hacer **comentarios** en XML se pone el texto entre las marcas:

```
<!-- y -->
```

El doble guión -- no debe aparecer en ninguna parte dentro del comentario hasta el cierre -->. En particular, se prohíbe el uso de tres guiones: --->

Los comentarios pueden aparecer:

- en cualquier parte en los datos de caracteres del documento
- antes o después del elemento raíz

Los comentarios no pueden aparecer:

- dentro de otro comentario
- dentro de una etiqueta

Las aplicaciones que leen y procesan documentos XML pueden o no pasar la información incluida en los comentarios.

Ejemplo del uso de comentarios:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Curso XML: Ejemplo de comentarios -->
<mensaje>
<![CDATA[
<parrafo>Esto es un curso de XML en el que
    las etiquetas <> son muy importantes
</parrafo>
]]>
<!-- Ejemplo uso de entidades predefinidas -->
<parrafo>Esto es un curso de XML en el que
    las etiquetas &lt; &gt; son muy importantes
</parrafo>
</mensaje>
```

## Tema 8. Ejemplos

### Ejemplo 1

A continuación, se muestra un ejemplo XML en el que se organiza y estructura la información de una agenda. Es decir, representaremos los contactos de nuestra agenda a través de la información de: nombre, dirección, empresa en la que trabaja, etc...

Antes de ver la solución te proponemos que dediques un par de minutos a pensar en cómo podría ser.

Solución propuesta:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Ejemplo que codifica en un fichero XML los contactos de una agenda -->
<mis_contactos>
  <contacto>
    <nombre> Juan Martin</nombre>
    <direccion>
      <calle> La plata </calle>
      <localidad> Madrid</localidad>
      <cp> 28080</cp>
      <pais> España </pais>
    </direccion>
    <telefono> 911234567 </telefono>
    <mail direccion="info@empresa.com"/>
    <empresa> Soluciones Globales</empresa>
  </contacto>

  <contacto>
    <nombre> Maria Pérez </nombre>
    <direccion>
      <calle> Avenida La Viña </calle>
      <localidad> Galicia</localidad>
      <cp>36001 </cp>
      <pais> España </pais>
    </direccion>
    <telefono> 981666666 </telefono>
    <mail direccion="info@ara.com"/>
    <empresa>Ara</empresa>
  </contacto>
</mis_contactos>
```

### Ejemplo 2

A continuación, se pretende estructurar en un documento XML la información relativa a los alumnos que acuden a una academia: nombre, dirección, si tienen o no domiciliado el pago, el curso, etc.

Solución propuesta:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Ejemplo que codifica en un fichero XML los contactos de una agenda -->
<academia>
```

```

<alumno>
  <nombre> Juan Martin</nombre>
  <direccion>
    <calle> La plata    </calle>
    <localidad> Madrid</localidad>
    <cp> 28080</cp>
    <pais> España </pais>
  </direccion>
  <telefono> 911234567 </telefono>
  <curso> Programacion VB NET </curso>
  <domicialiacion referencia="no"/>
</alumno>

<alumno>
  <nombre> concha Perez</nombre>
  <direccion>
    <calle> El oro    </calle>
    <localidad> Madrid</localidad>
    <cp> 28080</cp>
    <pais> España </pais>
  </direccion>
  <telefono> 911234567 </telefono>
  <curso> Programacion VB NET </curso>
  <domicialiacion referencia="1111-34-45-677777777"/>
</alumno>

<alumno>
  <nombre> Maria Trujillo</nombre>
  <direccion>
    <calle> Las Viñas    </calle>
    <localidad> Madrid</localidad>
    <cp> 28080</cp>
    <pais> España </pais>
  </direccion>
  <telefono> 911234567 </telefono>
  <curso> Ingles </curso>
  <domicialiacion referencia="no"/>
</alumno>
</academia>

```

### Ejemplo 3

Este ejemplo estructura la información de la descripción de un curso en un documento XML, a través del título del curso, de los módulos y temas que lo componen.

Solución propuesta:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Ejemplo que codifica en un fichero XML los contactos de una agenda -->
<curso>
  <titulo> Curso XML </titulo>
  <modulo>
    <titulo_mod>Modulo 1. Introduccion a XML</titulo_mod>
    <objetivos> Conocer qué es XML y para qué se utiliza </objetivos>
    <temas>

      <tema id="1"> Introduccion </tema>
      <tema id="2"> Ejemplos </tema>
      <tema id="3"> Tu Turno </tema>
      <tema id="4"> Recuerda Que.. </tema>
    </temas>
  </modulo>
</curso>

```



```
</modulo>
<modulo>
  <titulo_mod>Modulo 2. Estructura de un documento XML</titulo_mod>
  <objetivos> Conocer la sintaxis de XML </objetivos>
  <temas>

    <tema id="1"> Dcoumentos XML bien formados</tema>
    <tema id="2"> Partes de un documento XML </tema>
    <tema id="3"> Elementos </tema>
    <tema id="4"> Atributos </tema>
    <tema id="5"> Entidades predefinidas </tema>
    <tema id="6"> Secciones CDATA </tema>
    <tema id="7"> Comentarios </tema>
    <tema id="8"> Ejemplos </tema>
    <tema id="9"> Tu Turno </tema>
    <tema id="10"> Recuerda Que..</tema>

  </temas>

</modulo>
</curso>
```

## Tema 9. Recuerda Que...

Para que un documento XML esté correctamente escrito requiere de dos condiciones:

- Ser un documento válido.
- Ser un documento bien formado.

Los documentos XML deben tener una estructura jerárquica con lo que respecta a las etiquetas que delimitan sus elementos. Esto significa que los elementos deben estar correctamente anidados y que los elementos no se pueden solapar entre ellos.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<biblioteca>
  <libro id="Juv-010">
    <titulo> PORTA COELI: LA ORDEN DE SANTA CECLINA </titulo>
    <autor> SUSANA VALLEJO </autor>
    <categoria> Juvenil </categoria>
  </libro>
  <libro id="Int-011">
    <titulo> EL FUEGO </titulo>
    <autor> KATHERINE NEVILLE </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Int-211">
    <titulo> LA RESURRECCION MAYA </titulo>
    <autor> STEVE ALTEN </autor>
    <categoria> Literatura negra y de intriga </categoria>
  </libro>
  <libro id="Juv-917">
    <titulo> CREPUSCULO </titulo>
    <autor> STEPHENIE MEYER</autor>
    <categoria> Juvenil </categoria>
  </libro>
</biblioteca>
```

**Atributo**

**Elemento**

Por elemento entendemos el conjunto de fragmentos que componen la etiqueta. XML permite complementar a un elemento con atributos que matizan su significado.

Las entidades predefinidas son marcas XML que se utilizan para representar estos caracteres. El XML especifica cinco entidades predefinidas:

- ✂ & para el &
- ✂ &lt; para el <
- ✂ &gt; para el >
- ✂ &apos; para el '
- ✂ &quot; para el "

Otra solución, para escribir los caracteres especiales que ya tienen un significado propio para el procesador de XML (como el símbolo <, >, ""...) que no sean las entidades predefinidas es el uso de secciones CDATA. Es otra construcción en XML que permite especificar datos, utilizando cualquier carácter, especial o no, sin que se interprete como marcado XML.

Para hacer **comentarios** en XML se pone el texto entre las marcas:

```
<!-- y -->
```