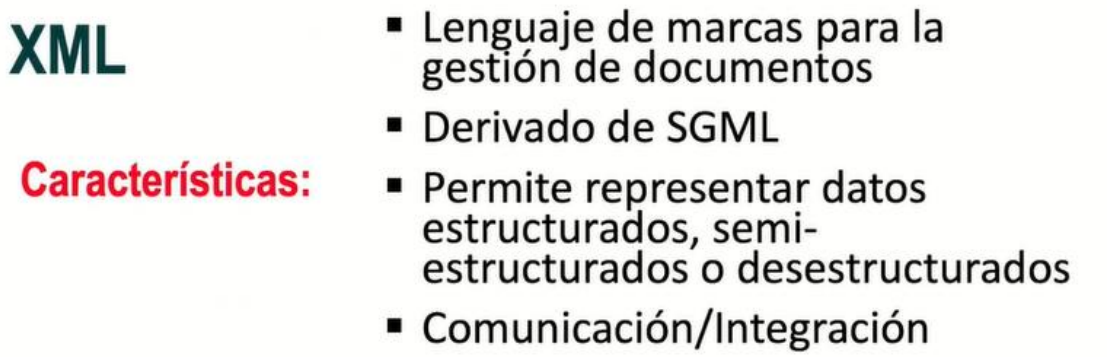
1. ¿Qué es XML?



XML es el lenguaje básico en las BBDD documentales y en la gestión normalizada de documentos, por lo que conocer su uso, sintaxis, y sus aplicaciones, es de utilidad a la hora de trabajar con sistemas de almacenamiento NO-SQL, o con sistemas de almacenamiento de documentos no normalizados.

Además es muy útil a la hora de intercambiar información entre distintos dispositivos, lo que hace que sea un lenguaje bastante utilizado y reconocido en la actualidad

Los documentos XML en definitiva son documentos de lenguajes de marcas, donde hay texto normal y etiquetas (marcas) que permiten clasificar dicho texto indicando su significado.

Las etiquetas en XML se deciden a voluntad, no hay una serie de etiquetas que se pueden utilizar. De hecho la función de XML es definir tipos de documentos etiquetados. Ejemplo de XML:

```
<persona>
        <nombre>
                Pepe
        </nombre>
        <apellido>
                Sánchez
        </apellido>
</persona>
```

El código es similar a HTML, sólo que las etiquetas se deciden según nos interese. Pero el funcionamiento el mismo:

- Las etiquetas tienen cierre que se indica con el signo / antes del nombre de la etiqueta.
- Las etiquetas afectan al texto (y otras etiquetas) que están entre la apertura y el cierre.

Por otro lado XML llama "**elemento**" a las etiquetas; mejor dicho, elemento sería tanto la etiqueta como lo que contiene.

En cualquier caso las normas son:

- Las etiquetas sirven para indicar elementos. El nombre de la etiqueta se indica entre los símbolos < y >.
- Las etiquetas se cierran indicando </ seguido del nombre de la etiqueta.
- XML distingue entre mayúsculas y minúsculas (en esto se diferencia de HTML) siendo buena práctica escribir las etiquetas en minúsculas.
- Se pueden espaciar y tabular las etiquetas a voluntad. Es buena práctica que un elemento interno a otro aparezca en el código con una sangría mayor (por eso en el ejemplo anterior ***nombre***, que está dentro de ***persona***, aparece sangrado).
- Los comentarios en el código se inician con los símbolos **<!--** y terminan con **-->.**
- Según la **W3C**, el texto en un documento XML debe de estar codificado en **Unicode** (normalmente **UTF-8**).

1.1. Estructura de un documento XML

Los documentos XML se dividen en:

- **Prólogo**. Se trata de la primera zona del documento y sirve para describir qué tipo de documento es. Es similar al apartado **head** de **HTML**. Puede contener
  - **Declaración del documento**, que permite indicar el tipo de documento XML que es.
  - **Instrucciones** para el procesado del documento
  - **Comentarios**
  - **Indicación del documento DTD** o el esquema para la validación.

- **Elemento raíz**. Todo el contenido del documento debe de estar incluido en el llamado elemento raíz, se trata de un elemento obligatorio que se abre tras el prólogo y se debe cerrar justo al final. De este modo cualquier elemento está dentro del elemento raíz. Contiene:
  - **Más elementos**
  - **Atributos**
  - **Texto normal**
  - **Entidades**
  - **Comentarios**

1.2. REGLAS PARA LOS NOMBRES

En XML los elementos, atributos,…. tienen nombre, el cual debe cumplir estas reglas:

- En XML se distingue entre mayúsculas y minúsculas, por lo que hay que tener cuidado al utilizar el nombre desde otro punto del documento.
- Deben comenzar por una letra, y después le seguirán más letras, números o el signo de subrayado o guion bajo.

1.3. ELEMENTOS DEL PRÓLOGO

- **Declaración XML:**
  Se trata de la primera línea de un documento XML e indica el tipo de documento XML que es (y así poder validar el mismo). En realidad es opcional, pero es muy recomendable.
  Es: **<?xml version**="1.0" **encoding=**"UTF-8"**?>**
  Indica la versión XML del documento y la codificación (utf-8 es la habitual).

- **Instrucciones de procesamiento:**
  Un documento XML puede incluir instrucciones de este tipo para indicar un documento para validar el XML, darle formato,… u otras funciones.

  Por ejemplo: **<?xml-stylesheet type**="text/xsl" **href**="stylesheet.xsl"**?>**

  Esta instrucción asocia un documento **xsl** al documento XML para poder darle un formato de salida (para especificar la forma en la que los datos se muestran por pantalla por ejemplo).

1.4. ELEMENTOS

Son la base del documento XML. Sirven para dar significado al texto o a otros elementos o también para definir relaciones entre distintos elementos y datos. Hay una confusión entre lo que es un elemento y lo que es una etiqueta. En este caso por ejemplo:

- <nombre>Pepe</nombre>
- <nombre> Es un etiqueta de apertura
- </nombre> Es una etiqueta de cierre
- <nombre>Pepe</nombre> Es un elemento
- Pepe es el contenido del elemento

El contenido de un elemento puede contener simplemente texto:

<descripción>
    Producto con precio rebajado debido a su escasa demanda
</descripción>

O puede contener otros elementos (o ambas cosas). En este, el elemento *persona* consta de un elemento *nombre* y otro *apellido*.

```
<persona>
        <nombre>Pepe</nombre>
        <apellido>Sánchez</apellido>
</persona>
```

Los elementos se deben abrir y cerrar con la etiqueta que sirve para definir el elemento; siempre se debe cerrar el último elemento que se abrió. Puede haber incluso elementos vacíos:

```
<casado></casado>
```

En este caso se pueden cerrar en la propia etiqueta de apertura:

```
<casado />
```

1.5. ATRIBUTOS

Se definen dentro de las etiquetas de apertura de los elementos. Se indica su nombre seguido del signo = y del valor (entre comillas) que se le da al atributo.

Ejemplo:

```
<persona complejidad="alta">
        <nombre>Pepe</nombre>
        <apellido>Sánchez</persona> ç
</apellido>
```

Un elemento puede contener varios atributos:

```
<persona privacidad="alta" tipo="autor">
        <nombre>Pepe</nombre>
        <apellido>Sánchez</persona>
</apellido>
```

1.6. TEXTO

El texto, como se comentó antes, está siempre entre una etiqueta de apertura y una de cierre. Eso significa que todo texto es parte de un elemento XML.

Se puede escribir cualquier carácter Unicode en el texto, pero no es válido utilizar caracteres que podrían dar lugar a confusión como los signos separadores < y > por ejemplo.

1.7. CDATA

Existe la posibilidad de marcar texto para que no sea procesado como parte de XML, eso se consigue colocándolo dentro de un elemento CDATA. Formato:
<! [CDATA [texto no procesable…**]]>**
Esto permite utilizar los caracteres < y > por ejemplo y no serán considerados como separadores de etiquetas.

```
<?xml version="1.0"?>
    <documento>
        <título>Prueba</título>
        <ejemplo>
            <![CDATA[ En HTML la negrita se escribe: <strong> ]]>
        </ejemplo>
    </documento>
```

En el ejemplo, los símbolos < y > no se toman como una etiqueta XML, sino como texto normal.

Otro uso de CDATA es colocar dentro de este elemento código de lenguajes de scripts como Javascript para que no sean interpretados como parte de XML.

1.8. ENTIDADES

Las entidades representan caracteres individuales. Se utilizan para poder representar caracteres especiales o bien caracteres inexistentes en el teclado habitual. Se trata de códigos que empiezan con el signo **&** al que sigue el nombre de la entidad o el número Unicode del carácter que deseamos representar.

En XML hay definidas cinco entidades:
> Símbolo **>**
< Símbolo **<**
& Símbolo **&**
" Símbolo **"**
' Símbolo **'**

También podemos representar caracteres mediante entidades con número. De modo que el ***ñ*** representa a la letra ñ (suponiendo que codificamos en Unicode, que es lo habitual). El número puede ser hexadecimal por ejemplo para la eñe de nuevo, sería **ñ**

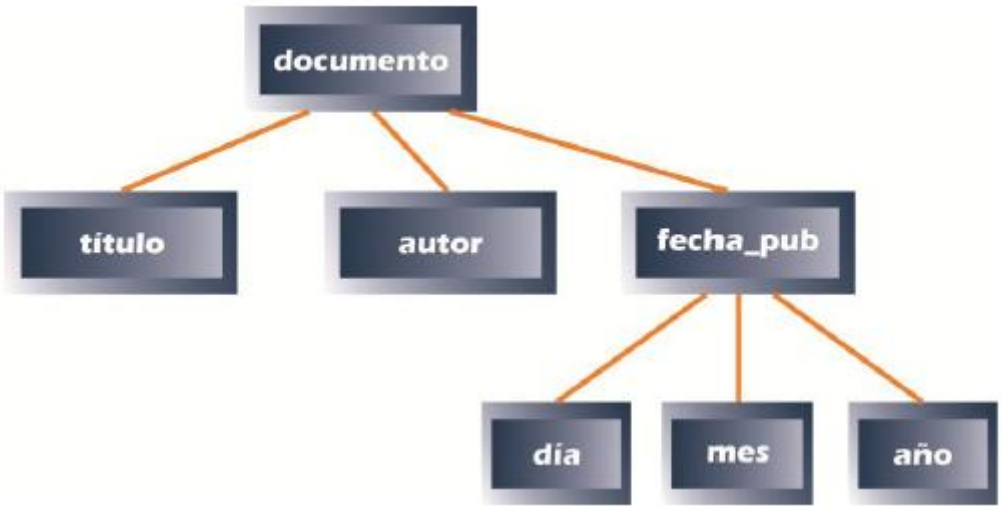
1.9. JERARQUÍA XML

Los elementos de un documento XML establecen una jerarquía que estructura el contenido del mismo. Esa jerarquía se puede representar en forma de árbol.
Así por ejemplo el archivo XML:

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

```
<documento>
        <título>Apuntes de XML</título>
         <autor>Pepe</autor>
        <fecha_pub>
                <día>18</día>
                <mes>Enero</mes>
                <año>2009</año>
        </fecha_pub>
</documento>
```

Gráficamente de forma jerárquica se podría expresar así:



2. XML BIEN FORMADO

Se habla de XML ***bien formado*** (**well formed**) cuando cumple reglas que facilitan su legibilidad y además permiten cumplir los objetivos formales de la edición de documentos XML.
Debemos tomar las reglas para producir XML bien formado como reglas obligadas, de hecho los analizadores (*parsers*) de XML indicarían error en caso de no cumplirlas. Es decir, un documento XML bien formado es un **documento analizable.**

2.1. REGLAS GENERALES

- Dentro del texto común no se pueden utilizar los símbolos de mayor (**>**), menor (**<**), ampersand (**&**) ni las comillas simples o dobles. Se deben de utilizar entidades o deben estar incluidos en una sección CDATA
- En principio en el texto normal, los símbolos de separación de caracteres como espacios en blanco, tabuladores y saltos de línea, no se tienen en cuenta. Es decir se usan al estilo de HTML. Pero sí es posible que sean

significativos en algunos elementos. En cualquier caso todos los caracteres escritos en el documento XML forman parte del mismo, será una cuestión posterior si se tienen en cuenta o no para presentar los datos del documento XML.

2.2. REGLAS PARA LOS ELEMENTOS

- Se deben cerrar primero las etiquetas de los últimos elementos abiertos.
- Las etiquetas son sensibles a mayúsculas y minúsculas.
- Todos los elementos poseen etiquetas de apertura y de cierre; es decir toda etiqueta que se abra se debe de cerrar. Si la etiqueta no tiene contenido, se debe cerrar en la propia etiqueta al estilo: **
**
- Todos los documentos XML deben de tener un único elemento raíz.
- Los nombres de los elementos comienzan con letras y pueden ir seguidos de letras, números, guiones o de puntos (los guiones y los puntos no son muy recomendables).
- Los nombres de los elementos no pueden comenzar con el texto *"xml"* (en mayúsculas ó minúsculas), ni con puntos ni caracteres de puntuación y tampoco pueden contener espacios.

2.3. REGLAS PARA LOS ATRIBUTOS

- Los nombres de atributos siguen las mismas normas que los nombres de los elementos.
- Los atributos sólo se pueden colocar en etiquetas de apertura y nunca en las de cierre.
- Los valores de los atributos deben ir entrecomillados (sin importar si se usan comillas simples o dobles).
- Todos los atributos deben de tener valor asociado. Es decir esta etiqueta (válida en HTML) no sería válida en XML:

  <hr noshade>

  Debería ser, por ejemplo, así:

  <hr noshade="noshade**">**

3. ESPACIOS DE NOMBRES

3.1. EL PROBLEMA DE LOS NOMBRES DE ELEMENTOS

Puede ocurrir que cuando se manejan documentos XML, que diferentes XML que tengamos, utilicen las mismas etiquetas. Aunque el contexto sería distinto, tendríamos un problema si manejamos ambos documentos con el mismo software, ya que el analizador, no sabría cómo manejar ambas etiquetas iguales. Los espacios de nombres (**namespacing** en inglés) evitan el problema indicando en cada etiqueta un código que sirve para indicar el contexto de cada etiqueta y así diferenciar las que son iguales. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
        <document>
                <title>Documento de prueba</title>
                <content>
                        <html>
                                <head> <title>Título HTML</title> </head>
                                <body> Texto del documento </body>
                        </html>
                </content>
                <author>Pepe</author>
        </document>
```

En el ejemplo anterior se usan etiquetas en inglés para el documento (algo muy habitual en el mundo empresarial) y eso hace que la etiqueta *title* se repita en contextos distintos, el primero es para poner un título genérico al documento (y es una etiqueta de la empresa en cuestión) y la segunda se corresponde a la etiqueta *title* del lenguaje HTML (o mejor XHTML).

La solución es anteponer al nombre de la etiqueta un nombre que indique el propietario de la misma, por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
        <document>
                <pepe.title>Documento de prueba</pepe.title>
                <content>
                        <html>
                                <head> <html.title>Titulo HTML</html.title>
                        </head>
                                <body> Texto del documento </body>
                         </html>
                </content>
                <author>Pepe</author>
        </document>
```

Ese prefijo diferenciador es el espacio de nombres al que pertenece la etiqueta, pero usado así tendríamos el problema de que con un sufijo tan corto, se podría repetir. Por ello una solución es indicar la URL de la entidad responsable de la etiqueta:

```
<?xml version="1.0" encoding="UTF-8"?>
 <www.pepe.net.document>
        <www.pepe.net.title> Documento de prueba </www.pepe.net.title>
        <www.pepe.net.content>
                <www.w3c.org.html>
                        <www.w3c.org.head>
                                <www.w3c.org.title> Título HTML
                                </www.w3c.org.title>
```

```
                    </www.w3c.org.head>
                    <www.w3c.org.body> Texto del documento
                    </www.w3c.org.body>
                </www.w3c.org.html>
            </www. pepe.net.content>
            <www. pepe.net.author> Pepe </www.pepe.net.author>
</document>
```

Pero el documento quedaría muy poco legible. Por ello se aplican espacios de nombres, de modo que se asigna una URL a un prefijo de etiqueta; de este modo cada vez que el documento se utiliza el prefijo, se sabe que se refiere a la URL indicada (las URLs son únicas).

3.2. USO DEL ATRIBUTO XMLNS

Todas las etiquetas en XML pueden hacer uso del atributo xmlns (xml namespacing) que permite asignar un espacio de nombres a un prefijo en el documento dentro del elemento en el que se usa el espacio de nombres. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
        <document xmlns:pp="http.//www.pepe.net/document"
        xmlns:html="http://www.w3c.org/html">
                <pp:title> Documento de prueba </pp:title>
                <pp:content>
                        <html:html>
                        <html:head>
                                <html:title> Título HTML </html:title>
                        </html:head>
                        <html:body> Texto del documento </html:body>
                </html:html>
                </pp:content>
                <pp:author> Pepe </pp:author>
        </document>
```

En el ejemplo se usa el prefijo ***pp*** para indicar etiquetas del espacio de nombres ***www.pepe.net/document*** y ***html*** para el espacio de nombres de HTML.

3.3. ESPACIO DE NOMBRES POR DEFECTO

En el caso de que las etiquetas, mayoritariamente, en un documento pertenezcan a un mismo espacio de nombres, lo lógico es indicar el espacio de nombres por defecto. Eso se hace sin indicar prefijo en el atributo **xmlns**. Ejemplo:

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

```
<document xmlns ="http.//www.pepe.net/document"
xmlns:html="http://www.w3c.org/html">
<title> Documento de prueba </title>
<content>
        <html:html>
                <html:head>
                        <html:title> Título HTML </html:title>
                </html:head>
                <html:body> Texto del documento </html:body>
        </html:html>
</content>
<author> Pepe </author>
</document>
```

Las etiquetas sin prefijo se entiende que pertenecen al espacio de nombres ***www.pepe.net/document***, para las del otro espacio se usa el prefijo.

3.4. USO DE ESPACIOS DE NOMBRES EN ETIQUETAS INTERIORES

El atributo **xmlns** no tiene por qué utilizarse en el elemento raíz, se puede posponer su declaración en el primer elemento que pertenezca al espacio de nombres deseado. Por ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
        <document xmlns ="http.//www.pepe.net/document">
                <title> Documento de prueba </title>
                <content>
                        <html:html xmlns:html="http://www.w3c.org/html">
                                <html:head>
                                        <html:title> Titulo HTML </html:title>
                                </html:head>
                                 <html:body> Texto del documento </html:body>
                        </html:html>
                </content>
                <author> Pepe </author>
        </document>
```

3.5. USO DE ESPACIOS POR DEFECTO EN ETIQUETAS INTERIORES

Un documento puede declarar espacios por defecto en etiquetas interiores lo que permite aún más versatilidad en los documentos. Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
        <document xmlns ="http.//www.pepe.net/document">
```

```
<!-- comienza el espacio de nombres de pepe.net -->
        <title> Documento de prueba </title>
        <content>
            <html xmlns="http://www.w3c.org/html">
            <!-- desde aquí el espacio ahora es el de html -->
                <head>
                    <title> Titulo HTML </title>
                </head>
                <body> Texto del documento </body>
            </html>
        <!-- fin espacio html, regresa el espacio pepe.net --> </content>
        <author> Pepe </author>
</document>
```

3.6. CANCELAR ESPACIOS POR DEFECTO

Si se usa el atributo xmlns="", entonces se está indicando (en el interior de la etiqueta en la que se use) que ese elemento y sus hijos no usan ningún espacio de nombres.