**LM TEMA 4: Lenguajes para el almacenamiento y transmisión de la información → XML:**

**XML** (Extensible Markup language) es para la gestión/almacenamiento de datos no para la presentación. En formato de texto plano. Es un metalenguaje (lenguaje que permite crear otros lenguajes) y está desarrollado por el W3C. Hay otros lenguajes que son alternativas al XML: JSON (JavaScript Object Notation).

XML (XML → fichero.xml) indica un diseño/formato que permite almacenar/transmitir datos en texto plano, y está **compuesto** por:

* **Elementos:** <inicio> Datos/Elementos </inicio>. Y las etiquetas son propias, los nombres lo decide quien lo crea. Toda etiqueta que se abre, se cierra, sí o sí.
* **Atributos** → Par de valores. Su valor es obligatorio que vaya entre comillas (dobles o simples, para poder mezclar “ sd ‘ad‘ asda“) y los nombres no pueden llevar espacio (ni de los elementos)

**Características**:

* Se puede definir un conjunto ilimitado de etiquetas que proporciona el marco de trabajo para estructurar los datos.
* El XML es fácil de distribuir.
* e**X**tensible también (a parte de que es estándar) quiere decir que se pueden ampliar.
* Sigue una estructura concreta.
* Los datos van separados del proceso y de la presentación (con etiquetas). Y para mostrarlo de una manera determinada hay que aplicarle estilos en una hoja aparte (puedes aplicarle css pero no es aconsejable, es para HTML) con **XSL** o XSLT (propios para aplicar estilos a XML, la T significa que se puede transformar), que generan conversiones de documentos (a html, pdf, etc…)
* El contenido de los datos es **autodescriptivo**.
* Sigue una regla estándar → W3C, organización que lo revisa.

Usaremos → **XML Copy Editor** (aunque falla y para validarlo hay que cerrarlo y volverlo a abrir), se guarda en .xml. Ctrl + u → Acercar.

Inkscape → Guarda en XML.

**Reglas** XML:

* Todo documento XML tiene una y **solo una** **raiz**. (la raíz en este ejemplo de documento xml creado es la <libreria>).
* **Prólogo** → En la parte superior. No es obligatorio. Suele contener la versión de xml que se usa, y tiene que ser la **primera línea** (si la pones, obligatoriamente estará en la primera línea). También puede llevar la codificación que se va a usar y otros atributos. Para decirle si está relacionado con un documento externo hay que añadir otro atributo → standalone = “yes” (no está relacionado) o “no” (está relacionado con otro y hay que decir con quién está relacionado → <?xml-stylesheet type="text/xsl" href="estilo.xsl" ?>).
* **Cuerpo** → Lo demás.
* Los **atributos** se ponen en una etiqueta de inicio y puede ser una lista de atributos separados entre ellos por espacios. Pero no abusar de los atributos, se pone como elemento (etiqueta) si hay muchos. Lo que sea datos adicionales propios → atributos. Lo que sea datos a transmitir → Elementos.
* **Entidades** → Ya definidas o que define el usuario. Para representar un símbolo literal que no se interprete. Definidas → amp, gt , lt, quot. **< → &lt;** También se puede poner en decimal (&#60;) o hexadecimal. **> → &gt; & → amp;** Las entidades se ponen en los **elementos** no en los atributos (no se puede poner ahí). Se pueden definir entidades **personalizadas** en la parte superior (zona Doctype), es como un alias → !Doctype nombredeEtiquetaRaiz… [ <!ENTITY nombredelaEntidad “loqueRepresenta”> ]> Y ahora cambiará el nombre largo que representa por el “alias” en todo el documento cuando pongas **&**alias**;** .

Si cumple las reglas se dice que está **bien formado** (se comprueba dándole al V que hay al lado de la impresora en el XML Copy Editor). Tiene que estar bien formado. Si se abre con el explorador y se ve el código está bien formado, si no lo está, se ve solo texto y no todo.

La sesión que permite poner un contenido de cualquier tipo que **no** quiero que se analice ni se **procese** hay que ponerlo en el elemento **CDATA**(se quedará literal) → <![**CDATA**[ contenido ]]> Pero sí se mostrará en el navegador (a diferencia de un comentario) etc, solo que como si fuera un comentario y no comprueba si está bien formado ni nada.

Los **comentarios** solo se pueden poner en una línea aislada (no mezclados en atributos) y no se pueden poner dentro 2 guiones y no puede haber un comentario dentro de un comentario. Se escriben así→ <!-- comentario -->. Los comentarios no son un nodo y no se va a mostrar en el navegador, solo es para quien lo crea/usuarios. La excepción es XPA que sí los cuenta como un nodo, pero no es lo habitual que sea parte de la información.

El XML puede dar información acerca de lo que contiene con unas instrucciones especiales llamadas **instrucciones de procesamiento** para decirle a otros software que lo manejan (analizadores…) acerca de su contenido. Se ponen entre **<?** así → **<?** contenido **?>**

Hay instrucciones de procesamiento que sirven para indicarle que ejecute código script (Como PHP) → <? **php** contenido ? >. También hay para enlazar a hojas de estilos u otros documentos.

**Espacios de nombres:** Proporcionan un método simple para cualificar nombres de elementos y atributos. También se pueden asignar alias (con la forma xmlns: mio=”mio.Damian.org“)

Un espacio de nombres se declara con las siglas -> xmlns

xmlns=”mio.Damian.org“ o xmlns = ”mio.Damian.org“

El valor del espacio de nombre tiene formato raíz y no se comprueba. En la declaración del espacio de nombres se hace en el elemento raíz del documento y los elementos que queramos indicar que pertenecen a un espacio de nombre en concreto llevan delante el alias del espacio de nombre (ej. <mio:Nombre>).

El primer requisito que un XML debe cumplir es que este bien formado (cumple las reglas de sintaxis de un XML), debe cumplir las siguientes condiciones:

* Solo contiene caracteres Unicode válidos.
* Solo tiene uno y solo un elemento raíz.
* Los nombres de los elementos y los atributos no contienen espacios ni caracteres raros (aunque admite bastantes).
* El primer carácter de un nombre (elemento o atributo) puede ser una letra, dos puntos o subrayado.
* Los caracteres < > & solo se utilizan como inicio o marca de un elemento o una entidad.
* Cada etiqueta abierta debe cerrarse (aún sea una etiqueta vacía), coincidiendo la de apertura y cierre en todo (mayúsculas y minúsculas).
* La etiqueta de cierre no puede llevar ningún alias (NO es el caso de los espacios de nombre, pues todo se convierte en etiqueta).
* No existen etiquetas con dos atributos iguales.
* Todos los atributos deben tener un valor asignado y siempre entre comillas.

**Documentos válidos:** Es válido cuando cumple unas determinadas reglas que define el usuario (el validador). El cual valida todo desde los elementos hasta los datos (contenido de los elementos).

El validados puede ir implícito en el documento (en la zona del DOCTYPE) o en otro archivo externo (DTD o Schema).