# APUNTES XSLT

## Para poder trabajar con XSLT

**Instalar en VSCODE:**

* **Live Server.** Para poder emular el contenido de las páginas que se van transformando a partir de xmls.
* **XML.** Para poder tener las ayudas en escritura de código XML.
* **XSLT.** Lo mismo pero para XSLT.

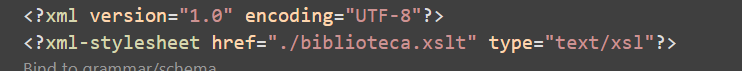
## VINCULANDO XSLT Y XML

Se genera el archivo que será la plantilla. Tienen extensión xslt. En VS code se genera lo siguiente:



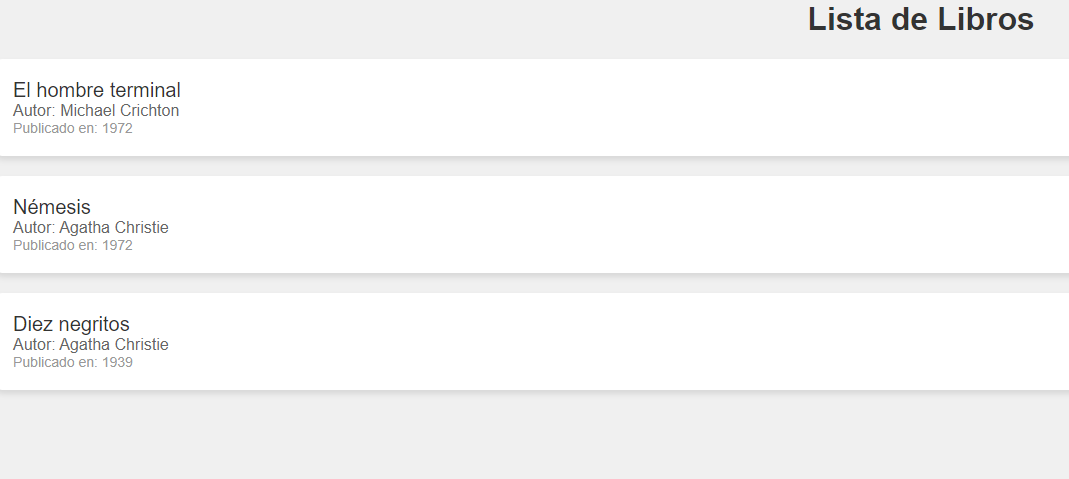
No importa coger el esquema de un archivo que sepamos que funciona. Este es el esqueleto básico para que se pueda realizar la transformación.

En el fichero xml escribimos la vinculación:



**La etiqueta xml-stylesheet identifica una plantilla xslt.**

Esto hará que el xml pase a tener un aspecto:



En caso de tener una plantilla vacía. El transformador de archivos va recorriendo todos los datos y se queda con el contenido de las etiquetas. Evita los atributos y no los recoge.

## MOSTRANDO DATOS DE UNA MANERA CONCRETA

**Etiqueta xsl:template match =”/”**



Usando en el atributo match la barra indicamos que queremos mostrar todo el árbol de datos que hay dentro del XML. El atributo match hace referencia a elementos usando el lenguaje XPATH,

**Etiqueta xsl:value-of select=”/biblioteca”**

Usando la etiqueta xsl:value-of select 🡪 Podemos seleccionar mediante una expresión XPATH los elementos que se van a renderizar en la página.

DENTRO DEL SELECT 🡪 SIEMPRE HAY UNA RUTA XPATH.

Ojo: Si usamos una ruta xpath muestra sólo el primer resultado de acceder a la misma.



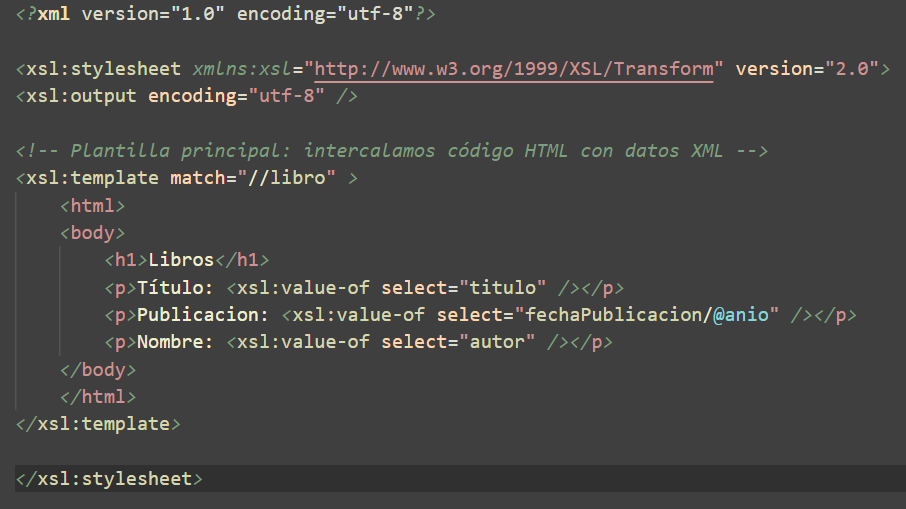
Este código selecciona al primer autor del primer libro de la biblioteca.

## USANDO OTRO CÓDIGO:

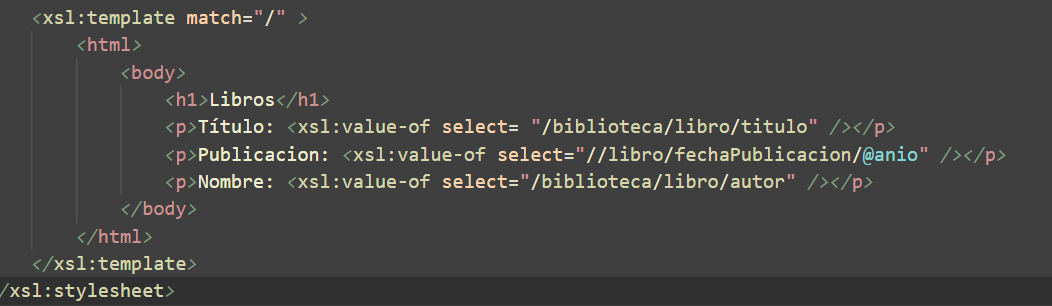
Se pueden usar etiquetas html intercaladas a la hora de crear las plantillas.



## TRABAJANDO CON UNA PLANTILLA I

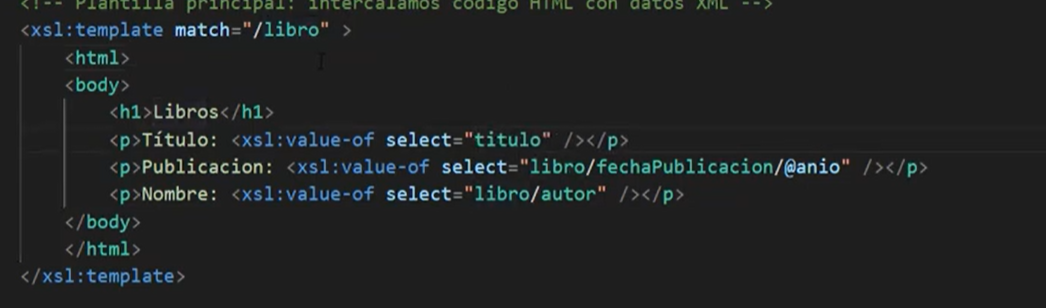


En la imagen ejemplo de plantilla de xslt. Con template match abrimos el lugar dónde se va a generar la plantilla. Match nos permite elegir la parte del árbol del xml que se va a representar. En este caso se elige la parte de libro. Se puede hacer diferente y utilizar rutas completas:



En este fragmento se accede únicamente a un fragmento de código y se accede únicamente al primer valor.

**Trabajando con rutas relativas y una absoluta en el match:**



La selección de los datos cambia en función de lo que se vaya poniendo. En el caso que se ve en la imagen detectamos tres fallos:

* Estamos trabajando con rutas relativas. En el caso de publicación la palabra “libro” sobra.
* En el caso de nombre, la palabra libró también va a sobrar.
* En el caso del match 🡪 Es una ruta absoluta se debe colocar: //libro.

Haciendo esto podemos seleccionar todos los elementos libro de la biblioteca. Si elegimos biblioteca, elegimos sólo un bloque de información. XLT por cómo está pensado si seleccionas un bloque de código te brindará únicamente el primer resultado de la búsqueda que realices. Es decir, selecciono biblioteca ( 1 elemento) y selecciono libro pues muestra sólo el primer libro del bloque de código.

## TRABAJANDO CON UNA PLANTILLA 2

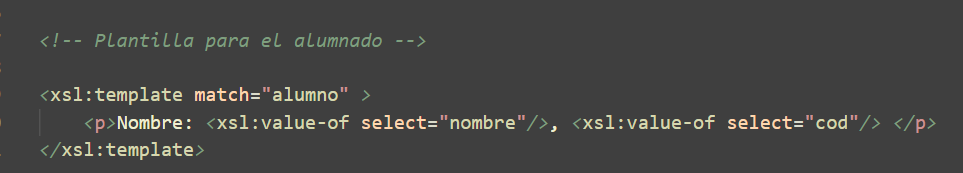
## TRABAJANDO CON VARIAS PLANTILLAS - APPLYTEMPLATES

Con XMLs complejos es interesante aplicar varias plantillas. Este primer punto explica el siguiente concepto:



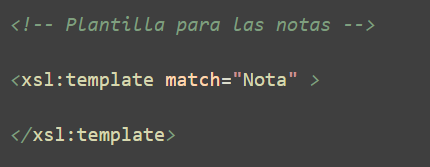
Esta etiqueta va a seleccionar todo el xml y va a permitir renderizar todas las plantillas que haya posteriormente en el xml.

Una vez hecho esto podemos pasar a crear la plantilla:



Usando match seleccionamos el tipo de nodo que queremos representar. En este caso seleccionamos todos los nodos alumno, vamos a imprimir su nombre y su código de alumno asociado.

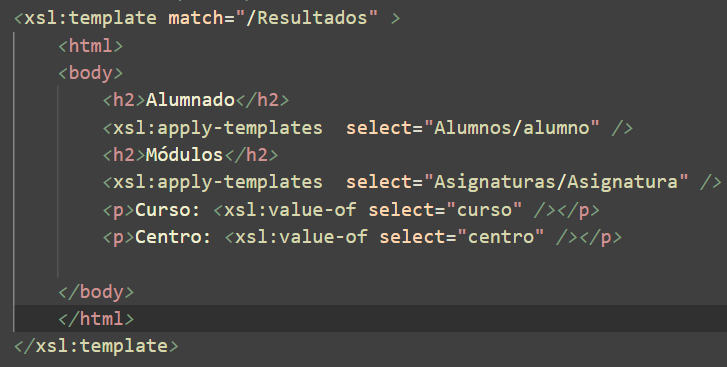
Es posible que queramos ocultar nodos en concreto. Para ello realizamos lo siguiente:



Si no renderizamos nada el contenido de los nodos nota no se mostrará.

### **USANDO EL SELECT**

Ahora vamos a ver el efecto de aplicar el select a la hora de aplicarlo. Lo mejor explicar con un ejemplo:



Usando el select somos capaces de hacer varias cosas:

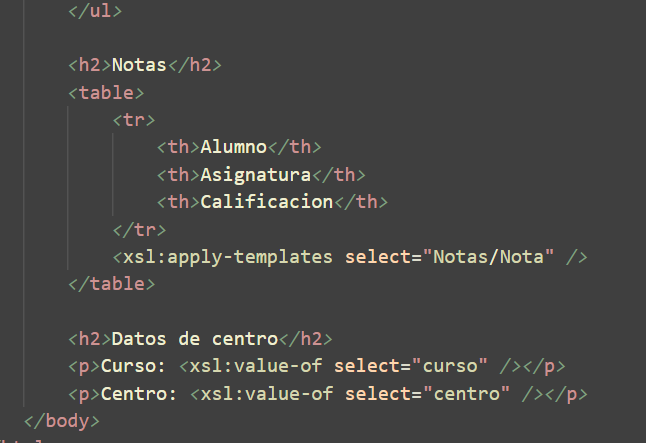
* Seleccionar unos nodos concretos. Al seleccionar el nodo alumno y el nodo asignatura podemos mostrar la información ahí contenida únicamente.
* Al usar select además podemos saltarnos el orden natural del xml y mostrar nodos que van después en el xml antes de lo que deberían.

En el ejemplo además se incorporan líneas de código de html entre medias para poder visualizar de mejor manera el resultado final.

### INTERCALAR HTML

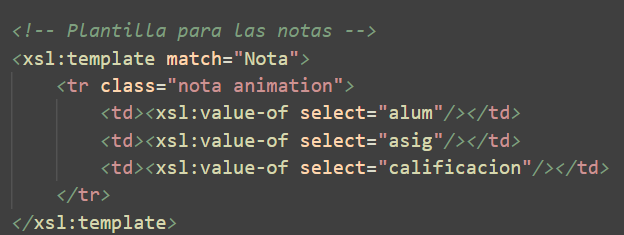
Las plantillas xslt dejan intercalar código HTML para optimizar el renderizado de la información que se va representando desde el xml.

Veamos un ejemplo:



En la imagen superior vemos dos tipos de renderizaciones:

* **TABLA**: En la tabla tenemos la parte que no queremos que se renderice todas las veces que se hace uso de la plantilla (la cabecera), después tenemos el template que va generando las líneas de la tabla. OJO: Son rutas relativas en el select. Debemos hacer coincidir la ruta del nodo que se muestra aquí con lo que aparece en el template en cuestión:



En la imagen vemos como la plantilla selecciona los nodos NOTA y coincidirá con la imagen vista anteriormente.

* **El renderizado normal**. Podemos hacer renderizados de datos generales para poder mostrar directamente.

### ORDENAR ELEMENTOS: