



LLENGUATGES DE MARQUES I SISTEMES DE GESTIÓ DE LA INFORMACIÓ

CURS 2018-2019

Gemma Simó Diego
gemma.simo.ies@gmail.com

Tema 6:

Conversió i Adaptació de Documents XML (XPATH, XSLT)

P.- INTRODUCCIÓ:

Fins ara hem vist:

- Que són els arxius XML
- Com crear arxius XML
- Com fer que els documents XML estiguin ben formats
- Com validar arxius XML (DTD, XML SCHEMA)

Que ens falta?

- Com manipular arxius XML
- Com transformar arxius XML

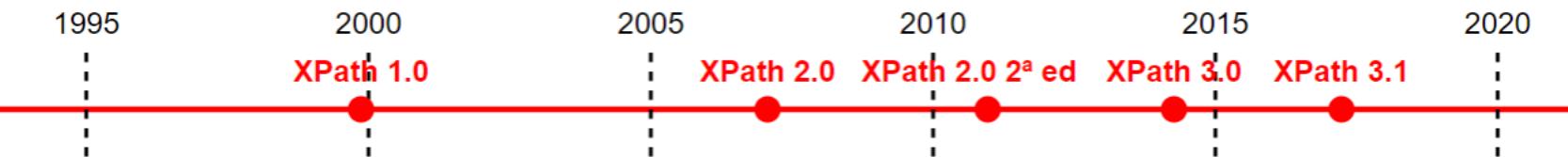
P.- INTRODUCCIÓ:

- **XPath:** Llenguatge de rutes XML, és un llenguatge per tal d'accedir a les diferents parts d'un document XML. Amb aquest llenguatge es poden recórrer i processar els documents XML.
- **XSLT:** eXtensible Stylesheet Language Transformation. Ens permet definir el mode per tal de transformar un document XML en un altre document.

2.- XPATH

XPath és un llenguatge que permet seleccionar nodes d'un document XML i calcular valors a partir del seu contingut. Existeixen varies versions de XPath aprovades per W3C.

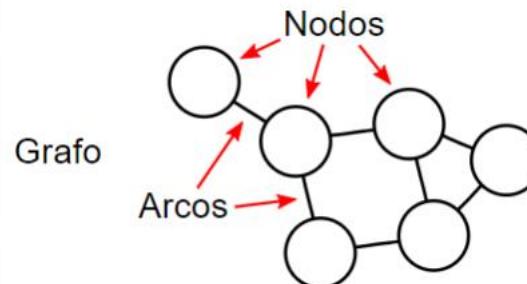
- noviembre de 1999: [XML Path Language \(XPath\) 1.0](#)
- enero de 2007: [XML Path Language \(XPath\) 2.0](#)
- diciembre de 2010: [XML Path Language \(XPath\) 2.0 \(2º edición\)](#)
- abril de 2014: [XML Path Language \(XPath\) 3.0](#)
- marzo de 2017: [XML Path Language \(XPath\) 3.1](#)



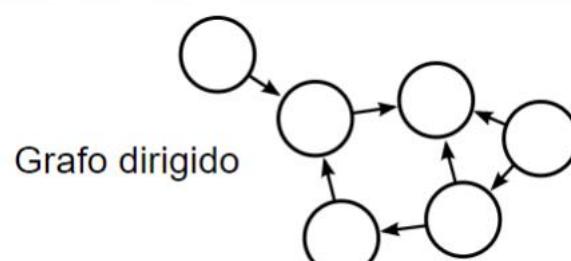
2.1- XPath: Arbre del document

XPath considera un document XML com un arbre de nodes.

- Un **graf o grafo** és un conjunt d'objectes anomenats nodes o vèrtex units per enllaços anomenats arcs o arestes.

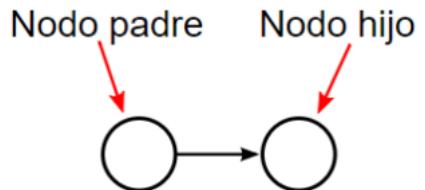


- Un **graf dirigit** és un graf en el que les arestes tenen direcció.

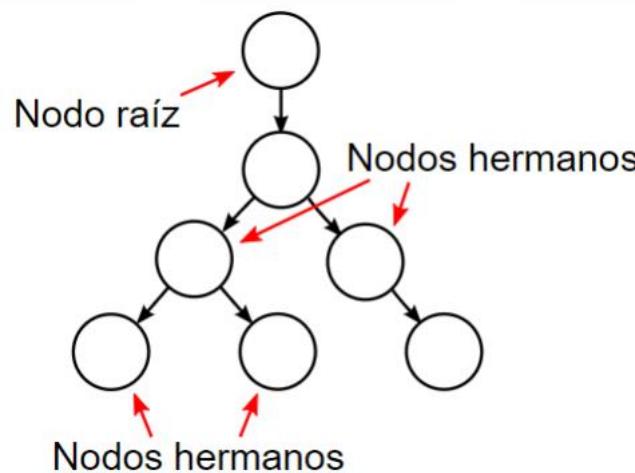


2.1- XPath: Arbre del document

- Quan 2 nodes estan units per un arc amb direcció, el node pare és el node del que parteix l'aresta i el node fill és el node al que arriba l'aresta.

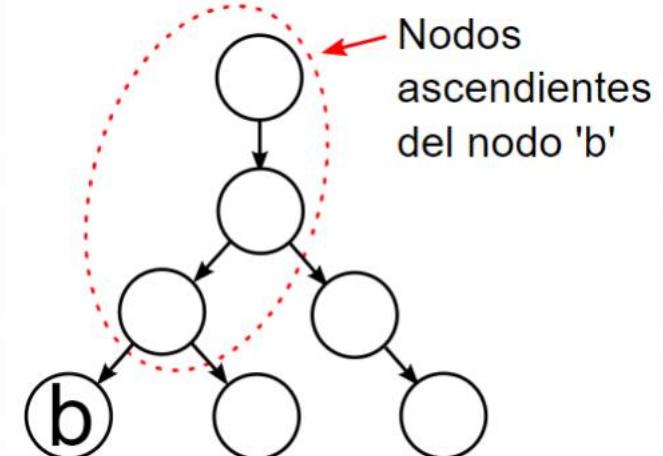
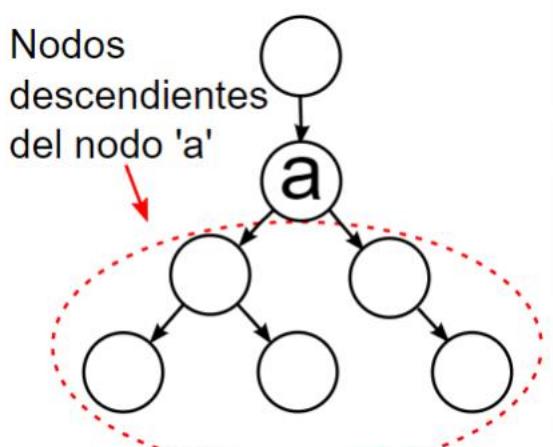


- El **nodo root o arrel** d'un arbre dirigit és l'únic node sense pare. Els **nodos germanos** són els nodes que tenen el mateix pare.



2.1- XPath: Arbre del document

- Els **nodes descendents** d'un node són tots els nodes als quals s'arriba des d'un node: els fills, els fills dels fills, etc.
- Els nodes ascendents d'un node són tots els nodes dels que un node és descendant: el pare, el pare del pare, etc.



2.2- XPath: Tipus de nodes

Un document XML pot representar-se com un arbre dirigit. Però en XPath no sols els elements són nodes, en realitat hi ha 7 tipus de nodes:

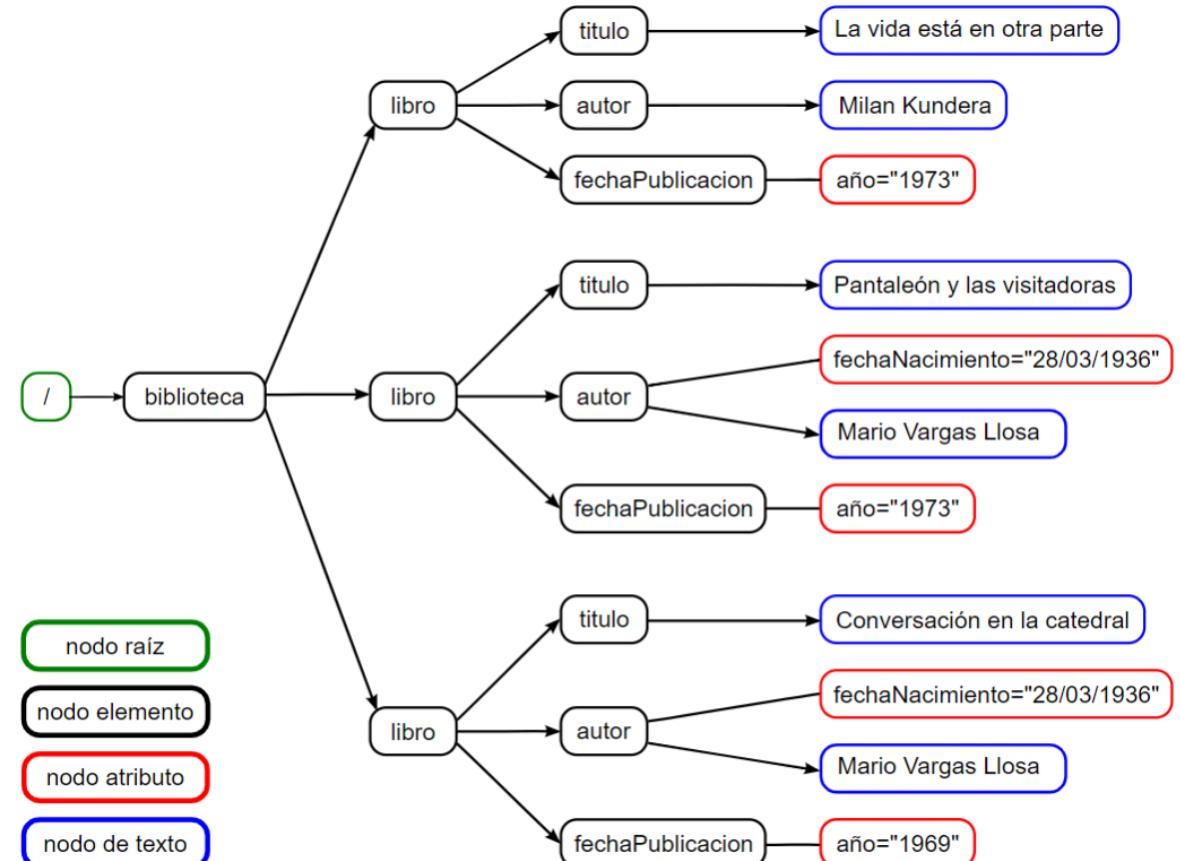
- Root o arrel
- Element
- Atribut
- Text (contingut d'un element)
- Comentari
- Instrucció de processament
- Espacio de nombres

Nota: La declaració DOCTYPE no es considera node.

2.2- XPath: Tipus de nodes

Per exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```



2.3- XPath: Sintaxis de les expressions de XPath

- Una **expresió XPath** és una cadena de text que representa el recorregut en l'arbre del document. Les expressions més simples es pareixen a les rutes dels arxius de Windows o en la shell de GNU/Linux.
- **Avaluat una expresió XPath** és cercar si hi ha nodes en el document que s'ajusten al recorregut definit en l'expressió. El resultat de l'avaluació són tots els nodes que s'ajusten a aquesta expressió. Per poder avaluar una expressió XPath el document ha d'estar ben format.

2.3- XPath: Sintaxis de les expressions de XPath

Les expressions XPath es poden escriure de 2 formes diferents:

- **sintaxis abreviada**: més compacta i fàcil de llegir
- **sintaxis completa**: més llarga però amb més opcions disponibles.

Les expressions XPath es poden dividir en passos de cerca. Cada pas de cerca es pot dividir al mateix temps en 3 parts:

- **eix**: selecciona nodes element o atribut basant-se en els seus noms.
- **predicat**: restringeix la selecció de l'eix a que els nodes compleixin certes condicions.
- **selecció de nodes**: dels nodes seleccionats per l'eix i predicat, selecciona els elements, el text que contenen ambdós.

2.3- XPath: Sintaxis abreviada

L'eix ens permet seleccionar un subconjunt de nodes del document i correspon a recorreguts en l'arbre del document. Els nodes element s'indiquen mitjançant el seu nom. Els nodes atribut s'indiquen mitjançant @ i el nom de l'atribut.

- /: si està al principi de l'expressió indica el node root, sinó indica "fill".
Ha d'anar seguida del nom d'un element.
- //: indica "descendent" (fills, fills dels fills, etc.).
- ../: indica l'element pare
- |: permet indicar diversos recorreguts

2.3- XPath: Sintaxis abreviada

- 1) Si pose /biblioteca/libro/autor que em retornarà?
- 2) Si pose /autor que em retornarà?
- 3) Si pose /biblioteca/autor que em retornarà?
- 4) Si pose /biblioteca/libro/autor/@fechaNacimiento que em retornarà?
- 5) Si pose /biblioteca/libro/@fechaNacimiento que em retornarà?

/biblioteca/libro/@fechaNacimiento que em

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose /biblioteca//autor que em retornarà?
- 2) Si pose //autor que em retornarà?
- 3) Si pose //autor//libro que em retornarà?
- 4) Si pose //@año que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose /biblioteca/libro/autor/@fechaNacimiento/.. que em retornarà?
- 2) Si pose //@fechaNacimiento/../../ que em retornarà?
- 3) Si pose //autor|//titulo que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

El **predicat** s'escriu entre corxetes a continuació de l'eix. Si l'eix ha seleccionat uns nodes, el predicat permet restringir aquesta selecció als que compleixen determinades condicions.

[@atribut]: selecciona els elements que tenen l'atribut indicat.

[número]: si hi ha diversos resultat selecciona un d'ells per ordre d'aparició; **last()** selecciona l'últim d'ellos

[condició]: selecciona els nodes que compleixin la condició.

Els predicats permeten definir condicions sobre els valors dels atributs.

2.3- XPath: Sintaxis abreviada

En les condicions es poden utilitzar els operadors següents:

- **operadors lògics:** and, or, not()
- **operadors aritmètics:** +, -, *, div, mod
- **operadors de comparació:** =, !=, <, >, <=, >=

Les comparacions es poden fer entre valors de nodes i atributs o amb cadenes de text o numèriques.

- En el cas de les cadenes de text deuen estar envoltades per cometes simples o dobles.
- En el cas de cadenes numèriques, les cometes són optatives.

2.3- XPath: Sintaxis abreviada

- La condició pot utilitzar el valor d'un atribut (utilitzant @) o el text que conté l'element.
- Per a fer referència al propi valor de l'element seleccionat s'utilitza el punt (.).
- Un predicat pot contenir condicions compostes.
- Es poden escriure diversos predicats seguits, cadascun dels quals restringeix els resultats de l'anterior, com si estiguessin encadenats per l'operació lògica and.

2.3- XPath: Sintaxis abreviada

- 1) Si pose //autor[@fechaNacimiento] que em retornarà?
- 2) Si pose //libro[1] que em retornarà?
- 3) Si pose //libro[last()] que em retornarà?
- 4) Si pose //libro[last()-1] que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose //fechaPublicacion[@año>1970] que em retornarà?
- 2) Si pose //libro[autor="Mario Vargas Llosa"] que em retornarà?
- 3) Si pose //año[.>1970] que em retornarà?
- 4) Si pose //autor[.= "Mario Vargas Llosa"] que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose //libro[autor="Mario Vargas Llosa" and fechaPublicacion/@año="1973"] que em retornarà?
- 2) Si pose //libro[autor="Mario Vargas Llosa" or fechaPublicacion/@año="1973"] que em retornarà?
- 3) Si pose //libro[autor="Mario Vargas Llosa"] [fechaPublicacion/@año="1973"] que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

Selecció de nodes (node test) : La selecció de nodes s'escriu a continuació de l'eix i del predicat. Si l'eix i el predicat han seleccionat uns nodes, la selecció de nodes indica amb quina part d'aquests nodes ens quedem.

- **/node()**: selecciona tots els fills (elements o text) del node.
- **//node()**: selecciona tots els descendents (elements o text) del node.
- **/text()**: selecciona únicament el text que conté el node.
- **//text()**: selecciona únicament el text que conté el node i els seus descendents.
- **/***: selecciona tots els fills (sols elements) del node.
- **//*[@**: selecciona tots els descendents (sols elements) del node.
- **/@***: selecciona tots els atributs del node.
- **//@***: selecciona tots els atributs dels descendents del node.

2.3- XPath: Sintaxis abreviada

- 1) Si pose //libro/node () que em retornarà?
- 2) Si pose //autor/node () que em retornarà?
- 3) Si pose //libro//node () que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose //autor/text () que em retornarà?
- 2) Si pose //libro/text () que em retornarà?
- 3) Si pose //libro//text () que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose /biblioteca/* que em retornarà?
- 2) Si pose //autor/* que em retornarà?
- 3) Si pose /biblioteca///* que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3- XPath: Sintaxis abreviada

- 1) Si pose //@* que em retornarà?
- 2) Si pose //libro/@* que em retornarà?
- 3) Si pose //autor/@* que em retornarà?

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
    <libro>
        <titulo>La vida está en otra parte</titulo>
        <autor>Milan Kundera</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Pantaleón y las visitadoras</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1973"/>
    </libro>
    <libro>
        <titulo>Conversación en la catedral</titulo>
        <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
        <fechaPublicacion año="1969"/>
    </libro>
</biblioteca>
```

2.3.1- XPath: Passes consecutives

Una expressió XPath pot contenir diverses passes de búsqueda consecutives. Cada passa inclourà el seu eix (i el predicat), i l'última passa inclourà una selecció de nodes. Cada pas de búsqueda treballa a partir dels nodes seleccionats pel cas de búsqueda anterior.

En l'exemple següent s'obtenen els títols dels llibres publicats després de 1970, mitjançant 2 passes de búsqueda.

- en el primer pas (`//fechaPublicacion[@año>1970]`) es seleccionen els elements `<fechaPublicacion>` que tinguin un atribut any superior a 1970.
- en el segon pas (`/../titulo`), es seleccionen primer els elements pare (`/..`) de `<fechaPublibacion>` seleccionats en el primer pas de búsqueda (és a dir, elements `<libro>`) i a continuació els seus subelements `<titulo>`.

2.3.1- XPath: Passes consecutives

Per tant, ho cercaríem de la següent forma:

```
//fechaPublicacion[@año>1970]/../titulo
```

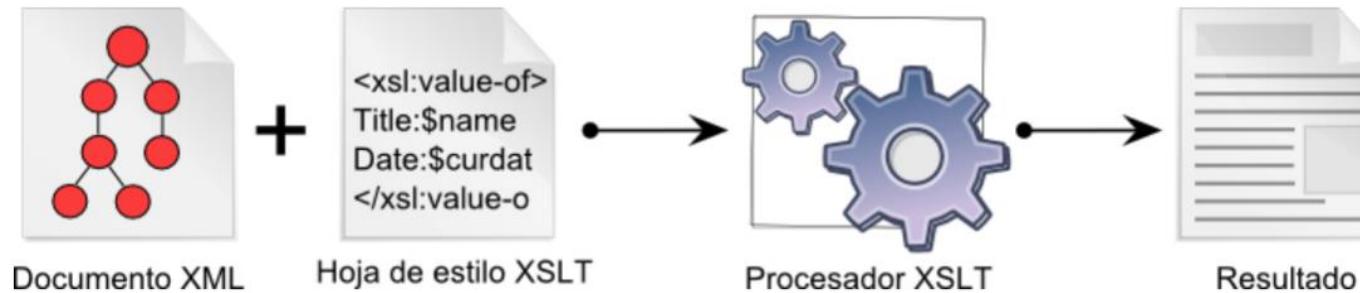
i obtindriem:

```
<titulo>La vida está en otra parte</titulo>
<titulo>Pantaleón y las visitadoras</titulo>
```

Hi ha moltes formes diferents per tal d'arribar al mateix resultat!

3- Transformacions XSLT

- XSLT (Transformacions XSL) és un llenguatge de programació declaratiu que permet generar documents a partir de documents XML:



- El document XML és el document inicial a partir del qual es va a generar el resultat.
- La fulla d'estil XSLT és el document que conté el codi font del programa, és a dir, les regles de transformació que s'aplicaran al document inicial.
- El processador XSLT és el programa d'ordinador que aplica al document inicial les regles de transformació incloses en la fulla d'estil XSLT i genera el document final.
- El resultat de l'execució del programa és un nou document.

3- Trasformacions XSLT

XSLT s'utilitza per a obtenir a partir d'un document XML altres documents (XML o no).

A un document XML se li poden aplicar diferents fulles d'estil XSLT per a obtenir diferents resultats i una mateixa fulla d'estil XSLT es pot aplicar a diferents documents XML.

El llenguatge XSLT està normalitzat per el W3C, que ha publicat 3 versions:

- novembre de 1999: [XSLT 1.0](#)
- gener de 2007: [XSLT 2.0](#)
- juny de 2017: [XSLT 3.0](#)



3.1- Fulles d'estil XSLT

XSLT és un llenguatge declaratiu. Per això les fulles d'estil XSLT no s'escriuen com una seqüència d'instruccions, sinó com una col·lecció de plantilles (template rules). Cada plantilla estableix com es transforma un determinat element (definit mitjançant expressions XPath). La transformació del document es realitza de la següent manera:

- El processador analitza el document i construeix l'arbre del document.
- El processador recorre l'arbre del document des del node arrel.
- En cada node recorregut, el processador aplica o no alguna plantilla:
 - Si a un node no se li pot aplicar ninguna plantilla, el seu contingut s'inclou en el document final. A continuació el processador recorre els seus nodes fills.
 - Si a un node se li pot aplicar una plantilla, se li aplica. La plantilla pot generar text que s'inclourà en el document final. En principi, el processador no recorrerà els seus elements fills, a no ser que estiga indicat en la plantilla.
 - Quan el processador ha recorregut tot l'arbre ha acabat la transformació.

3.1- Fulles d'estil XSLT

Una fulla d'estil XSLT és un document XML que conté al menys aquestes etiquetes:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
</xsl:stylesheet>
```

Aquestes etiquetes són:

- la declaració xml `<?xml>`, pròpia de qualsevol document XML.
- la instrucció `<xsl:stylesheet>` és l'etiqueta arrel de la fulla d'estils, els seus atributs indiquen la versió i l'espai de noms corresponent.

Dins de la instrucció `<xsl:stylesheet>` podem trobar els elements de nivell alt i les plantilles. Com l'exemple següent:

3.1- Fulles d'estil XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:output method="xml" indent="yes"/>

  <xsl:template match="/">
    </xsl:template>
```

Aquestes etiquetas són

- l'element de nivell alt `<xsl:output>` indica el tipus de sortida produït.
- l'instrucció `<xsl:template>` és una plantilla.
 - L'atribut **match** indica els elements afectats per la plantilla i conté una expressió XPath.
 - El contingut de l'instrucció defineix la transformació a aplicar (si la instrucció no conté res, com en l'exemple, substituirà el node per res, és a dir, eliminarà el node, encara que conservarà el text que conté l'element).

3.1- Fulles d'estil XSLT

Quan s'aplica una plantilla a un node, en principi no es recorren els nodes descendents. Per a indicar que sí volem recórrer els nodes descendents i aplicar-los les plantilles que els corresponen, hi ha que utilitzar l'instrucció `<xsl:apply-templates />`, com en el següent exemple:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

    <xsl:template match="/">
        <xsl:apply-templates />
    </xsl:template>

    <xsl:template match="elemento">
    </xsl:template>

</xsl:stylesheet>
```

3.2- Enllaçar documents XML amb fulles XSLT

Es pot associar de forma permanent una fulla d'estils XSLT a un document XML mitjançant la instrucció de processament `<?xmlstylesheet ... ?>`.

La instrucció de processament `<?xmlstylesheet ... ?>` va al principi del document, després de la declaració XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xmlstylesheet type="text/xsl" href="ejemplo.xsl"?>
```

Quan es visualitze en un navegador web un document XML enllaçat amb una fulla d'estils XSLT, els navegadors mostren el resultat de la transformació. Hi ha diferències entre els diferents navegadors.

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

**Si no tinguessim plantilles:
el xslt és d'aquesta forma:**

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
</xsl:stylesheet>
```

**ens tornaria tot el text del
document:**

```
<?xml version="1.0" encoding="UTF-8"?>
  La vida está en otra parte
  Milan Kundera
  Pantaleón y las visitadoras
  Mario Vargas Llosa
  Conversación en la catedral
  Mario Vargas Llosa
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>  
<biblioteca>  
  <libro>  
    <titulo>La vida está en otra parte</titulo>  
    <autor>Milan Kundera</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Pantaleón y las visitadoras</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Conversación en la catedral</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1969"/>  
  </libro>  
</biblioteca>
```

Si hi ha una plantilla buida:
el resultat inclou el contingut dels nodes
`<titulo>` ja que no hi ha regla per a ells, però
no els de `<autor>` perquè aquesta està
buida.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="1.0">  
  <xsl:template match="autor">  
  </xsl:template>  
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>  
  La vida está en otra parte  
  Pantaleón y las visitadoras  
  Conversación en la catedral
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>  
<biblioteca>  
  <libro>  
    <titulo>La vida está en otra parte</titulo>  
    <autor>Milan Kundera</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Pantaleón y las visitadoras</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Conversación en la catedral</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1969"/>  
  </libro>  
</biblioteca>
```

Si la plantilla buida se li aplica al node arrel:

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="1.0">  
  
  <xsl:template match="/">  
  </xsl:template>  
  
</xsl:stylesheet>
```

ens tornarà:

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

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

La instrucció `<xsl:value-of>` extrau el contingut del node seleccionat.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="libro">
    <xsl:value-of select="autor"/>
  </xsl:template>
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
Milan Kundera
Mario Vargas Llosa
Mario Vargas Llosa
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>  
<biblioteca>  
  <libro>  
    <titulo>La vida está en otra parte</titulo>  
    <autor>Milan Kundera</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Pantaleón y las visitadoras</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Conversación en la catedral</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1969"/>  
  </libro>  
</biblioteca>
```

En aquest exemple s'obtenen el títol i l'autor dels llibres, però uns a continuació dels altres. Els salts de línia es creen després de cada aplicació de la regla (és a dir, a cada llibre), però no en l'interior de la regla.

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="1.0">  
  <xsl:template match="libro">  
    <xsl:value-of select="titulo"/>  
    <xsl:value-of select="autor"/>  
  </xsl:template>  
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>  
  
La vida está en otra parteMilan Kundera  
Pantaleón y las visitadorasMario Vargas Llosa  
Conversación en la catedralMario Vargas Llosa
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

En aquest exemple els autors s'obtenen gràcies a la regla que extrau el contingut del node (el caràcter punt ".") fa referència al propi element) i els títols s'obtenen perquè no hi ha regles per a eixe node.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="autor">
    <xsl:value-of select="."/>
  </xsl:template>
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
  La vida está en otra parte
  Milan Kundera

  Pantaleón y las visitadoras
  Mario Vargas Llosa

  Conversación en la catedral
  Mario Vargas Llosa
```

3.3-Exemples plantilles XSLT

També es poden extreure els valors dels atributs utilitzant @.

```
<?xml version="1.0" encoding="UTF-8"?>  
<biblioteca>  
  <libro>  
    <titulo>La vida está en otra parte</titulo>  
    <autor>Milan Kundera</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Pantaleón y las visitadoras</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1973"/>  
  </libro>  
  <libro>  
    <titulo>Conversación en la catedral</titulo>  
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>  
    <fechaPublicacion año="1969"/>  
  </libro>  
</biblioteca>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet  
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
  version="1.0">  
  <xsl:template match="fechaPublicacion">  
    <xsl:value-of select="@año"/>  
  </xsl:template>  
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>  
  La vida está en otra parte  
  Milan Kundera  
  1973  
  
  Pantaleón y las visitadoras  
  Mario Vargas Llosa  
  1973  
  
  Conversación en la catedral  
  Mario Vargas Llosa  
  1969
```

3.3-Exemples plantilles XSLT

Generar text addicional:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
<p>Milan Kundera</p>
<p>Mario Vargas Llosa</p>
<p>Mario Vargas Llosa</p>
```

3.3-Exemples plantilles XSLT

Generar text addicional:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
    <p><xsl:value-of select="titulo"/></p>
  </xsl:template>
</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
<p>Milan Kundera</p>
<p>La vida está en otra parte</p>
<p>Mario Vargas Llosa</p>
<p>Pantaleón y las visitadoras</p>
<p>Mario Vargas Llosa</p>
<p>Conversación en la catedral</p>
```

3.3-Exemples plantilles XSLT

Generar text addicional:

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/> escribió
      <xsl:value-of select="titulo"/></p>
  </xsl:template>

</xsl:stylesheet>
```

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
<p>Milan Kundera escribió "La vida está en otra parte"</p>
<p>Mario Vargas Llosa escribió "Pantaleón y las visitadoras"</p>
<p>Mario Vargas Llosa escribió "Conversación en la catedral"</p>
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

**Aplicar regles a subnodes:
l'instrucció **<xsl:apply-templates>****

Fa que s'apliquen als subelements
les regles que els siguin aplicables.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
  <xsl:template match="/">
    <html>
      <h1>Autores</h1>
      <xsl:apply-templates />
    </html>
  </xsl:template>
  <xsl:template match="libro">
    <p><xsl:value-of select="autor"/></p>
  </xsl:template>
</xsl:stylesheet>
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

Aplicar regles a subnodes:
l'instrucció **<xsl:apply-templates>**

Fa que s'apliquen als subelements
les regles que els siguin aplicables.

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
<html><h1>Autores</h1>
  <p>Milan Kundera</p>
  <p>Mario Vargas Llosa</p>
  <p>Mario Vargas Llosa</p>
</html>
```

3.3-Exemples plantilles XSLT

```
<?xml version="1.0" encoding="UTF-8"?>
<biblioteca>
  <libro>
    <titulo>La vida está en otra parte</titulo>
    <autor>Milan Kundera</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Pantaleón y las visitadoras</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1973"/>
  </libro>
  <libro>
    <titulo>Conversación en la catedral</titulo>
    <autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>
    <fechaPublicacion año="1969"/>
  </libro>
</biblioteca>
```

Aplicar regles a subnodes:
l'instrucció **<xsl:apply-templates>**

Fa que s'apliquen als subelements
les regles que els siguin aplicables.

ens tornarà:

```
<?xml version="1.0" encoding="UTF-8"?>
<html><h1>Autores</h1>
  <p>Milan Kundera</p>
  <p>Mario Vargas Llosa</p>
  <p>Mario Vargas Llosa</p>
</html>
```

3.3-Exemples plantilles XSLT

Cerqueu que fan **<xsl:text>**, **<xsl:strip-space>** i **<xsl:attribute>** i poseu exemples d'aquestes instruccions dins d'un xslt i expliqueu quin resultat ens tornaria (centrant-nos amb l'xml que tenim d'exemple).