# UF2. NF2. ANNEXE: XPATH (NAVEGACIÓ PER DOCUMENTS XML)

Índex

1. INTRODUCCIÓ 2

1.1. Què és XPath? 3

1.2. Primer exemple 3

2. CONSULTES 4

2.1. Consultes relatives o absolutes 4

3. NODES 4

3.1. Relacions entre nodes 4

3.2. Node Arrel 5

3.3. Elements 5

3.4. Atributs 6

3.5. Text 6

3.6. Comentaris 6

3.7. Funcions 6

3.8. Sintaxis de XPath 8

3.9. XPath Axes 9

3.10. Filtres o predicats 11

3.11. Expressió for 11

3.12. Expressió if 12

4. Recursos 13

# INTRODUCCIÓ

XPath s'utilitza per navegar pels elements i atributs en un document XML.

XPath, processa el document XML, amb un parser i construeix un **arbre de nodes**. Aquest arbre comença amb l’element arrel i tots els elements que pengen d’ell i acaba en nodes fulla, que contenen dades o atributs.

**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>

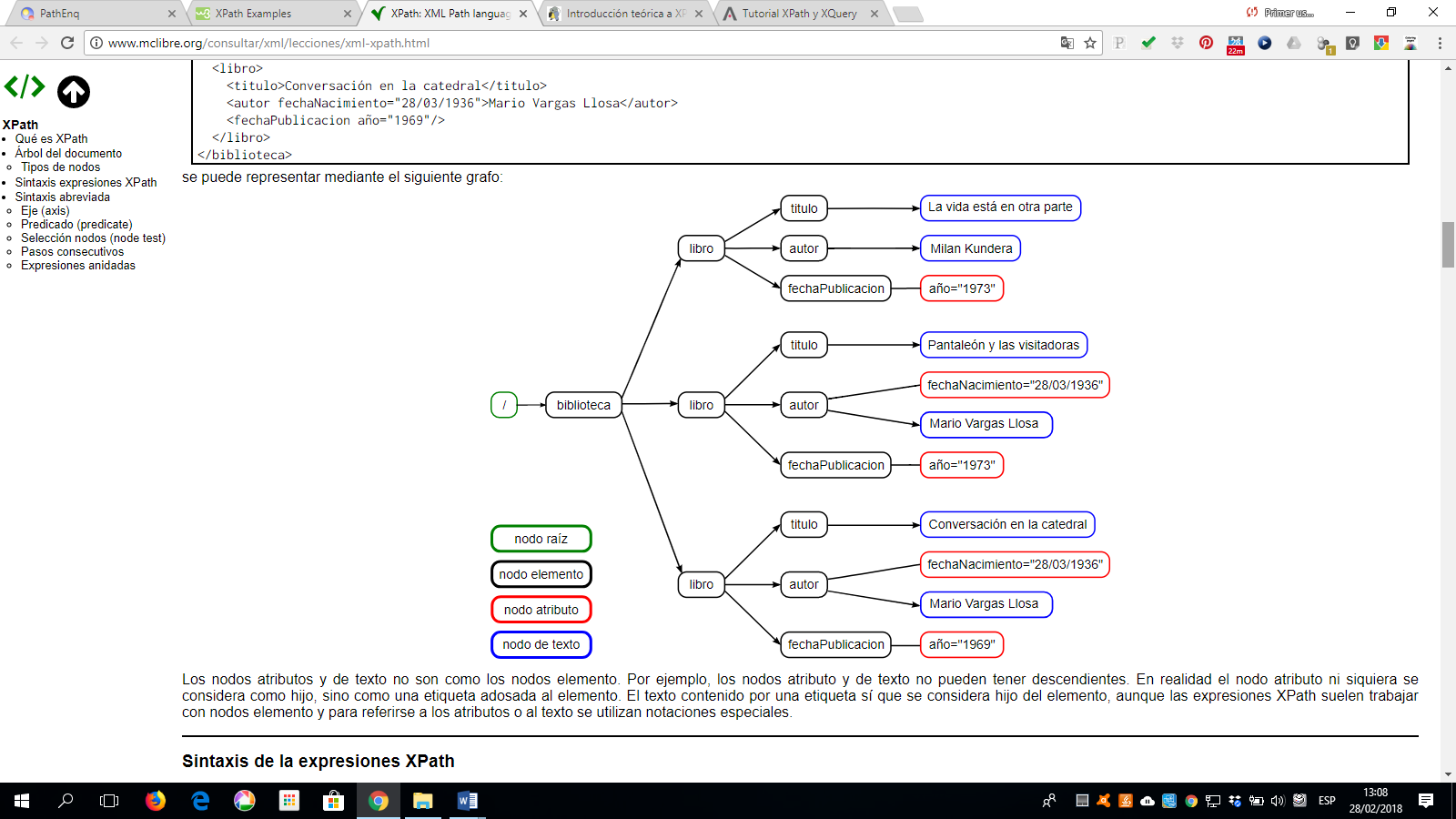
<autor fechaNacimiento="28/03/1936">Mario Vargas Llosa</autor>

<fechaPublicacion año="1969"/>

</libro>

</biblioteca>

**Arbre generat**



## Què és XPath?

* XPath és una sintaxis per definir parts d'un document XML.
* XPath usa expressions de ruta per navegar en documents XML.
* XPath conté una llibreria de funcions estàndard.
* XPath es un component de XSLT.
* XPath es una recomanació de W3C.

## Primer exemple

En aquesta unitat, treballarem amb [movies.xml](http://es.wikieducator.org/Usuario:Lmorillas/modulo_lenguajes_de_marcas/xml/xpath/ejercicios_movies).

Amb XPath aconseguirem contestar a preguntes com aquestes:

* Quin és el títol d'aquesta pel·lícula? /movies/movie/title
* Quants productors coneixem? count(/movies/movie/producer)
* Quins són els seus noms complets? //producer
* Quants directors té? count(/movies/movie/director)

# CONSULTES

Les consultes realitzades amb XPath tornen valors de diferents tipus:

* **Llistes de nodes**: si intentem obtenir, per exemple, la llista d'intèrprets de la pel·lícula.
* **Valors booleans**: si intentem saber si el document XML conté alguna crítica.
* **Cadenes**: si intentem obtenir el nom complet (nom i cognoms) del director de la pel·lícula.
* **Números**: si intentem obtenir el nombre d'intèrprets dels que es té notícia.

## Consultes relatives o absolutes

L'element if comprova si una certa condició booleana és veritable o falsa. Així, una instrucció condicional pot determinar si una acció es porta a terme o no.

Es diu que és absoluta si parteix del node arrel, és a dir, que el primer caràcter de la consulta és /. És el mateix que amb fitxer.

Exemple: Podem arribar a una crítica mitjançant:

* **Una ruta absoluta**: /movies/movie/comments/comment
* **Una ruta relativa:** comments/comment, a partir de movie.

# NODES

Per a XPath, tots els elements que componen un document XML són nodes, així existeixen diferents classes de nodes:

* **Arrel**: El node arrel sempe és /, del qual penja l'element arrel del document XML.
* **Instruccions de procés**: directives que permeten definir aspectes del document, així com declarar-lo com document XML.
* **Elements**: els blocs sobre els quals es dóna suport XML.
* **Atributs**: propietats dels elements.
* **Text**: els valors inclosos en l'interior dels elements, entre etiquetes d'obertura i tancament.
* **Comentaris**: també podem accedir al contingut de tots els comentaris.

## Relacions entre nodes

### Pare (Parent)

Cada element i atribut té un únic pare.

En el següent exemple, l'element “llibre” és el pare de títol, autor, any i preu:

<llibreria>

<llibre>

<titol>Rinya de gats</titol>

<autor>Eduardo Mendoza</autor>

<any>2010</any>

<preu>20.50</preu>

</llibre>

...

</llibre>

### Fill (Children)

Els elements nodes poden tenir zero, un o més fills.

En l'exemple anterior, els elements títol, autor, any i preu són tots fills de l'element llibre.

### Germans (Siblings)

Nodes que tenen el mateix pare.

En l'exemple anterior els elements títol, autor, any i preu són tots germans.

### Avantpassats (Ancestors)

El pare del pare del pare ... d'un node.

En l'exemple següent els avantpassats de l'element títol són els element llibre i element llibreria.

### Descendents (Descendants)

El fill d'un fill d'un fill... d'un node.

En l'exemple anterior els descendents de l'element llibreria són els elements llibre, títol, autor, any i preu.

## Node Arrel

Obtenir el node arrel és el procés més senzill, només hem de preguntar per la barra inclinada, operació que ens tornarà tot el document XML.

## Elements

Per seleccionar un determinat element només és necessari indicar la seva ruta. L'element més senzill és el node arrel.

Per exemple, si busquem el node “director” en el document XML de l'exemple, la seva ruta és: /movie/director

Una altra consulta també pot retornar un conjunt de nodes. Per exemple: /movie/actors/actor retorna tots els nodes d'intèrprets que apareguin al document.

## Atributs

Per mostrar el valor dels atributs d'un node hem d'accedir a l'element que ens interessi. Així, per accedir a l'any de la pel·lícula, la consulta és: [/movies/movie/@any](file:///C:/Users/teresa/movies/movie/@any)

La @ diferencia entre elements i atributs.

## Text

Per obtenir el text d'un determinat element podem utilitzar text(). Així, per obtenir el text d'un títol, la consulta és: /movies/movie/title/text()

## Comentaris

Per obtenir el valor d'un determinat comentari hem d'utilitzar comment(). Per obtenir el comentari del document és: /movies/movies/comments/comment

## Funcions

A més d'accedir als valors inclosos en un determinat document XML, també podem realitzar operacions a partir d'elements, com per exemple, concatenar cadenes o contar elements.

Exemple: Quants actors tenim en la pel·lícula?

count (/movies/movie/actors/actor)

Vegem les funcions més importants que podem utilitzar.

### Funcions booleanes

|  |  |
| --- | --- |
| **boolean()** | Pren un objecte com argument i retorna un valor booleà.   * Si l'argument és un número, retorna true si el número no és zero. * Si l'argument és un conjunt de nodes, retorna true si el conjunt de nodes no és buit. * Si l'argument és un string, retorna true si la cadena és no buida. |
| **false()** | No porta arguments i retorna false |
| **not()** | Porta com argument una expressió booleana i retorna true si l'expressió avaluada és false, i false si l'expressió és true. |
| **true()** | No porta arguments i retorna false |

### Funcions de conjunts de nodes

|  |  |
| --- | --- |
| **count()** | Pren com argument un conjunt de nodes i retorna el valor corresponent al nombre de nodes del conjunt de nodes. |
| **id()** | Pren com argument un string i retorna conjunt de nodes contenent qualsevol node que tingui un atribut de tipus ID igual que el valor de l'argument. |
| **last()** | Retorna un valor igual que l'últim node del conjunt. |
| **position()** | Retorna un número corresponent a la posició actual. |

### Funcions numèriques

|  |  |
| --- | --- |
| **ceiling()** | Pren un número com argument i retorna l'enter més petit que és més gran que l'argument. |
| **floor()** | Pren un número com argument i retorna el número enter més alt que es menor que l'argument. |
| **number()** | Pren com argument un string, booleà o un conjunt de nodes i:   * Si és un string i conté caràcters que són números, retorna aquests números. * Si és el booleà true retorna 1, i si és el booleà false retorna 0. |
| **round()** | Arrodoneix un número a l'enter més pròxim. |
| **sum()** | Pren com arguments un conjunt de nodes i retorna la suma del valor de cada node individual després de convertir el valor a numèric si és possible. |

### Funcions de cadena

|  |  |
| --- | --- |
| **concat()** | Pren com arguments dues o més cadenes i retorna la concatenació d'aquestes cadenes. |
| **contains()** | Pren com arguments dos strings i retorna true si la primera cadena conté a la segona. |
| **normalize-space()** | Pren una cadena com argument. Els espais són convertits a espais simples i després eliminats. |
| **starts-with()** | Pren com arguments dos strings i retorna true si el primer argument comença amb el segon argument. |
| **string()** | Pren un booleà, un conjunt de nodes o un número i retorna un valor string. |
| **string-length()** | Pren com argument un string i retorna la longitud del mateix. |

## Sintaxis de XPath

XPath usa expressions de ruta per accedir a un node o a un conjunt de nodes. XPath usa expressions de ruta per seleccionar nodes. El node és seleccionat mitjançant una sèrie de rutes o trajectòries. Les expressions de ruta més utilitzades són les següents:

|  |  |
| --- | --- |
| **Operador path** | **Descripció** |
| nodename | Selecciona tots els nodes fills del node nombrat |
| / | Selecciona des del node arrel |
| // | Selecciona nodes en el document des del node actual, no important on es troben (“recursive descendant”) |
| . | Selecciona el node actual |
| .. | Selecciona el pare del node actual |
| @ | Selecciona atributs |
| \* | “wildcard” (comodí que indica tot) |

Vegem amb el següent exemple com es seleccionen els nodes amb els operadors anteriors.

<empleats>

<empleat ID=”1234”>

<nom>Toni</nom>

<cognom>Magranet</cognom>

</empleat>

</empleats>

|  |  |
| --- | --- |
| **Exemple d'operador PATH** | **Explicació** |
| empleats | Selecciona tots els nodes fills de l'element empleats. |
| /empleats | Selecciona l'element arrel empleats. |
| empleats/empleat | Selecciona tots els elements empleat que són fills d'empleats. |
| //empleat | Selecciona tots els elements empleats no important on es trobin en el document. |
| empleats//nom | Selecciona l'element nom de qualsevol nivell per baix de l'element empleats. |
| //@ID | Selecciona tots els atributs anomenats com ID. |
| empleat/\* | Selecciona tot per sota de l'empleat. |
| empleat/@\* | Selecciona tots els atributs que hi ha per sota d'empleat. |

Utilitzant l'operador | en una expressió XPath podem seleccionar diverses rutes.

La taula següent mostra alguns exemples d'això:

|  |  |
| --- | --- |
| **Expressió Path** | **Resultat** |
| //empleat/nom | //empleat/cognom | Selecciona tots els elements nom i els elements cognom de tots els elements empleat. |
| //nom | //cognom | Selecciona tots els elements nom i tots els elements cognom en el document. |

## XPath Axes

Un axis defineix un conjunt de nodes en relació amb el node actual.

|  |  |
| --- | --- |
| **Nombre Axis** | **Resultat** |
| ancestor | Selecciona tots els avantpassats (ancestors) del node actual. |
| ancestor-or-self | Selecciona tots els avantpassats (ancestors) del node actual. I el node actual. |
| attribute | Selecciona tots els atributs del node actual. |
| child | Selecciona tots els fills del node actual. |
| descendant | Selecciona tots els descendents del node actual. |
| descendant-or-self | Selecciona tots els descendents del node actual i el node en si mateix. |
| following | Selecciona tot en el document després de l'etiqueta de tancament del node actual. |
| following-sibling | Selecciona tots els germans després del node actual. |
| parent | Selecciona el pare del node actual. |
| preceding | Selecciona tot en el document que es trobi abans de l'etiqueta abans de l'etiqueta d'apertura del node actual. |
| preceding-sibling | Selecciona tots els germans abans del node actual. |
| self | Selecciona el node actual. |

Una trajectòria de localització (location path) consisteix en:

* Un axis
* Un node-test
* Un predicat (opcional)

axis::node-test[predicate]

Vegem alguns exemples:

|  |  |
| --- | --- |
| **Exemple** | **Resultat** |
| child::empleat | Seleccionem tots els nodes empleats que són fills del node actual. |
| attribute::ID | Selecciona l'atribut ID del node actual. |
| child::\* | Selecciona tots els fills del node actual. |
| atribute::\* | Selecciona tots els atributs del node actual. |
| child::text() | Selecciona tots els nodes de tipus text del node actual. |
| child::node() | Selecciona tots els nodes fills del node actual. |
| descendant:: empleado | Selecciona tots els descendents d'empleats del node actual. |
| ancestor::empleado | Selecciona tots els avantpassats d'empleat del node actual. |
| ancestor-or-self::empleado | Selecciona tots els avantpassats d'empleat i el node actual si és un node empleat. |
| child::\*/child::apellido | Selecciona tots els nets cognom del node actual. |

## Filtres o predicats

Podem aplicar també filtres o predicats en la consulta de XPath. Van entre corchets. Podem usar operadors relacionals como:

= != > >= < <= ~=

Vegem diversos exemples:

/movie[producers/@producer~= 'Stephen\*']

També podem usar operadors aritmètics (+ - \* div mod)

/movie[(@minuts – 5)<(100\*0.05)]/producer

Vegem més exemples:

* **Presència d'elements fills**: empleat[salari]/nom, ens situem en l'elemento nom, per sota d'empleat que tinguin salari. Si tenim un element empleat que no tingui salari no el selecciona.
* **Valor dels elements fills**: empleat[salari>2500] ens situem en l'element empleat, el salari del qual sigui >2500.
* **Presència d'atributs**: empleat[@ID], selecciona els element empleat que tinguin un atribut ID.
* **Valor d'atributs:** empleat[@ID=”1234”] busquem l'element pel valor de l'atribut.

## Expressió for

Si has treballat amb llenguatges de programació, sabràs que el bucle "for" és una eina freqüent que existeix en la majoria dels llenguatges moderns i que en XPath és un component clau.

En la seva forma més senzilla, una expressió "for" serveix per recórrer una seqüència de valors, nodes fent servir una variable de bucle temporal que es defineix directament en l'expressió "for".

**Exemple**

for $ i in (1 to 10) return 2 \* $ i

Produirà la seqüència numèrica (2, 4, 6, 8, 10, 12, 14, 16, 18, 20) recorrent la seqüència original (1 to 10) usant la variable $ ii calculant 2 \* $ i per cada punt de dades de la seqüència.

Ara que sabem que qualsevol expressió XPath pot retornar una seqüència de nodes, també podem usar l'expressió "for" per recórrer aquests nodes i realitzar algunes operacions matemàtiques. Per exemple, per recórrer els conceptes d'una factora i calcular la quantitat corresponent per a aquest concepte, podem usar aquesta expressió:

for $ x in $ DB1 / sals / invoice / item return $ x / price \* $ x / quantity

En aquest cas, $ x serà la nostra variable de bucle que recorrerà cada concepte de la factura i aquesta expressió retornarà una seqüència que conté les quantitats per a cada concepte de la factura. Imaginem que tenim aquests dos conceptes en la font de dades de la factura:

<item>

<quantity> 5 </ quantity>

<price> 12.50 </ price>

</ item>

<item>

<quantity> 3 </ quantity>

<price> 27.85 </ price>

</ item>

Llavors l'expressió XPath anterior tornarà la seqüència (62.5 83.55) que són les quantitats de cada concepte de la factura.

## Expressió if

En els llenguatges de programació i d'expressions sol ser necessari decidir-se per una forma alternativa de calcular un resultat determinat, depenent de si es compleixen o no certs requisits. Per això és fonamental la instrucció if, que permet definir resultats alternatius depenent dels criteris de prova que vulgui utilitzar.

En XPath la instrucció if s'utilitza d'aquesta manera:

if (expressió-de-prova) then expression else expression

sent expressió-de-prova una expressió XPath lògica el resultat pot ser true o false.

Per exemple, en MobileTogether pot usar un comandament com aquest perquè el color del text d'una etiqueta o d'un camp d'edició depengui del valor numèric de les dades. Imagini que un element de la font de dades està associat amb un camp d'edició. Podem editar el camp d'edició i en la pestanya Propietats podem seleccionar la propietat Color del text. En lloc de definir el color amb el selector de colors, podem fer clic a la icona XPath de la barra d'eines i definir el color del text per mitjà d'una expressió XPath:

if ($ MT\_ControlValue> = 0) then "green" else "xarxa"

Aquí fem servir una de les variables integrades de MobileTogether per referir-nos al control actual i vam provar si el seu valor és més gran o igual que zero.

Depenent del resultat d'aquesta prova, el resultat serà "green" (verd) o "xarxa" (vermell) i el color del text canviarà depenent de les dades que introdueixi l'usuari en el control.

MobileTogether ofereix moltes variables integrades que comencen amb "$ MT\_" i que serveixen per a prendre decisions en funció de la mida de la pantalla del dispositiu mòbil, del sistema operatiu i de molts altres factors. Totes aquestes variables globals integrades estan en el quadre de diàleg "Variables globals" del menú Projecte de MobileTogether Designer.

# Recursos

* Validador online: <http://qutoric.com/xslt/analyser/xpathtool.html>
* [Exercicis resolts](http://www.mclibre.org/consultar/xml/ejercicios/xpath.html)
* <https://www.w3schools.com/xml/xpath_intro.asp>
* <https://ioc.xtec.cat/materials/FP/Materials/IC_S_INF/INF_IC_S_M04/web/html/WebContent/u3/a2/continguts.html> (apartar 2.4)