

UT7. Emmagatzament i consulta de documents XML

UT7. Emmagatzament i consulta de documents XML

7.1 Introducció a XPath

7.2 Arbres XPath

7.3 Navegació per l'arbre: Expressions

7.4 Atributs

7.5 Comodins

7.6 Eixos XPath

7.7 Condicions

7.8 Funcions

7.1 Introducció a XPath

XPath (XML Path language), és un llenguatge de descripció de rutes per a documents XML, desenvolupat per la W3C.

Ens permet seleccionar els nodes del document XML, i així poder manipular el contingut de les dades (de text, numèriques, etc).

La base del funcionament d'XPath és l'avaluació d'expressions. El resultat d'aquestes avaluacions poden ser de tipus booleà, numèriques un grup d'elements.

XSLT, es serveix de XPath per a la transformació de documents.

7.2 Arbre XPath

Arbre XPath

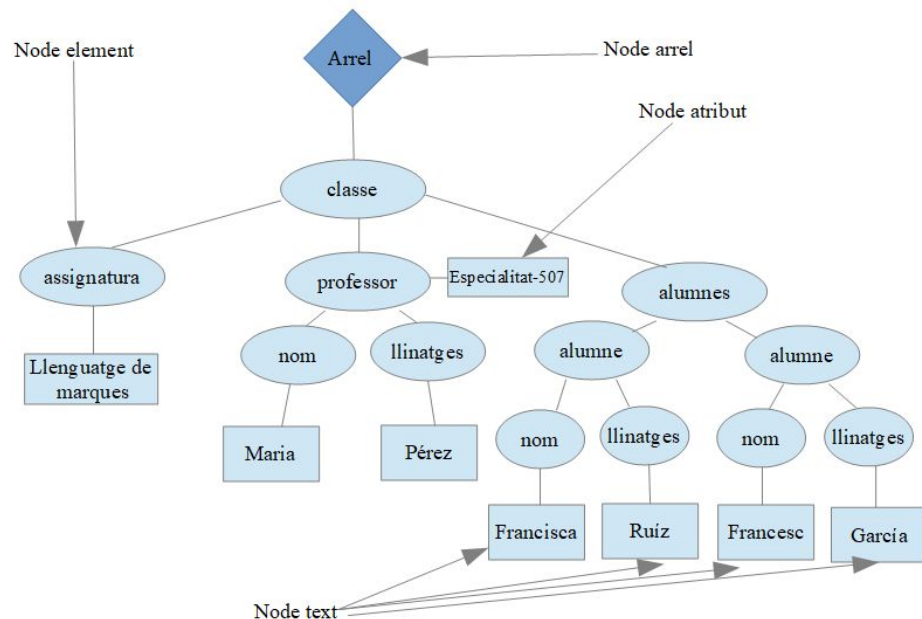
XPath tracta tots els documents XML des del punt de vista d'un arbre de nodes. L'arrel de l'arbre no es correspon amb l'arrel del document, sinó que és el propi document, i es representa amb el símbol “/”.

Igual que l'arbre del DOM, els nodes representen els elements, els atributs, els comentaris, i a més, a les instruccions de procés i també als espais de noms.

7.2 Arbre XPath

Exemple: Tenim el següent XML

```
<?xml version="1.0" ?>
<classe>
  <assignatura>Llenguatges de marques</assignatura>
  <professor Especialitat="507">
    <nom>Maria</nom>
    <llinatges>Pérez</llinatges>
  </professor>
  <alumnes>
    <alumne>
      <nom>Francisca</nom>
      <llinatges>Ruíz</llinatges>
    </alumne>
    <alumne>
      <nom>Francesc</nom>
      <llinatges>Garcia</llinatges>
    </alumne>
  </alumnes>
</classe>
```



En un arbre XPath, els atributs no són considerats nodes fills, sinó que són propietats del node que els conté.

7.3 Navegació per l'arbre: Expressions

Navegació per l'arbre: Expressions

Ja saben que la representació interna del document XML per XPath és un arbre. Per això, s'hi pot navegar especificant-hi camins d'una manera similar a com es fa en els directoris dels sistemes operatius.

Es pot dir que XPath utilitza expressions de ruta per a seleccionar nodes i conjunts de nodes en un document XML. Les expressions XPath poden utilitzar-se en JavaScript, Java, XML Schema, Python, C, C++, etc.

Una expressió és l'element base, que s'utilitza per a produir un resultat.

El més important en el moment de crear una expressió XPath, és saber el node en el qual està situat el procés (node de context), ja que és des d'aquest node, que s'avaluarà l'expressió. El node de context al principi és a l'arrel, però es va movent a mesura que es van avaluant les expressions.

7.3 Navegació per l'arbre: Expressions

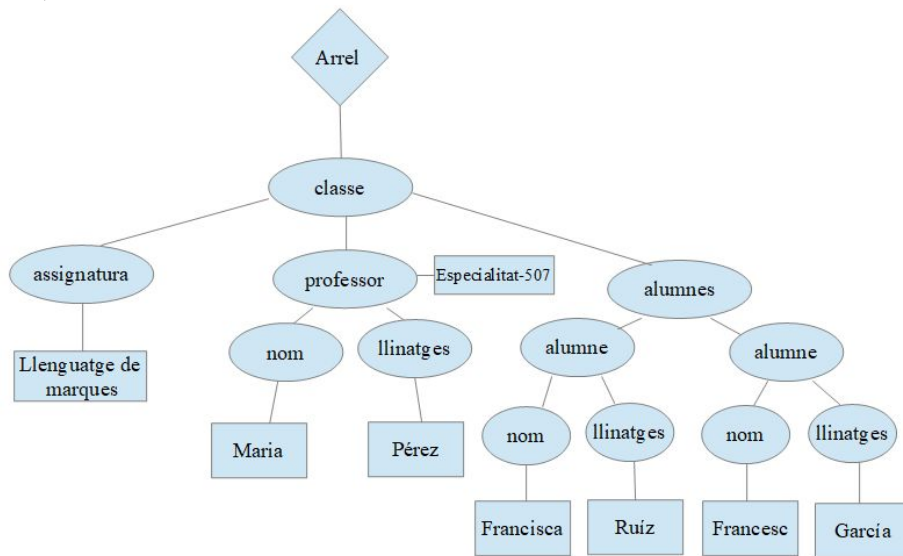
Podem expressar els camins XPath de dues maneres:

- **Camins absoluts:** Són camins que sempre comencen en l'arrel de l'arbre. Es poden identificar perquè el primer caràcter de l'expressió sempre serà l'arrel `"/`. No importa quin sigui el node de context si es fan servir camins absoluts, perquè el resultat sempre serà el mateix.
- **Camins relatius:** Parteixen des del node en el qual estem situats.

7.3 Navegació per l'arbre: Expressions

Exemple: Recuperem l'exemple vist al principi del capítol, l'XML és

```
?xml version="1.0" ?>
<classe>
  <assignatura>Llenguatges de marques</assignatura>
  <professor Especialitat="507">
    <nom>Maria</nom>
    <llinatges>Pérez</llinatges>
  </professor>
  <alumnes>
    <alumne>
      <nom>Francisca</nom>
      <llinatges>Ruíz</llinatges>
    </alumne>
    <alumne>
      <nom>Francesc</nom>
      <llinatges>Garcia</llinatges>
    </alumne>
  </alumnes>
</classe>
```



Es pot obtenir el node <nom> del professor, fent servir l'expressió XPath següent:

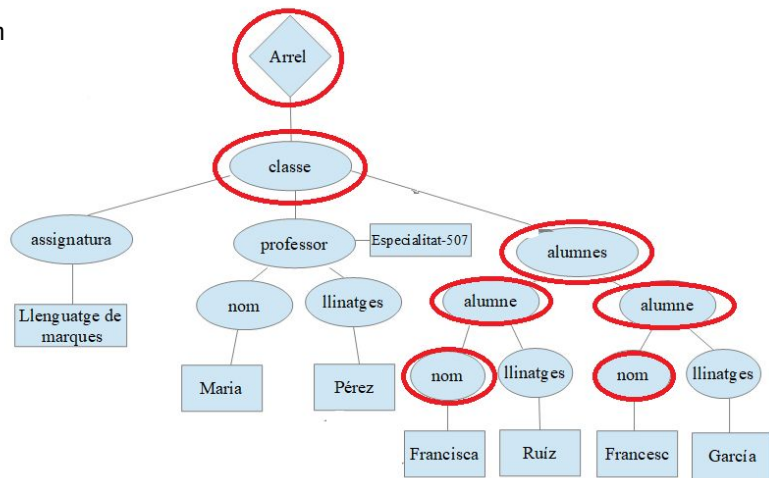
```
/classe/professor/nom
```

S'ha de tenir en compte que el resultat d'aquesta expressió no és només el contingut de l'element sinó tot l'element <nom>, el resultat seria: <nom>Maria</nom>

7.3 Navegació per l'arbre: Expressions

I també, molt important, sempre tenir en compte que l'expressió sempre intenta aconseguir el nombre màxim de camins correctes i, per tant, no necessàriament ha de retornar només un sol valor. Per exemple, si l'expressió per avaluar fos la següent: `/classe/alumnes/alumne/nom`, XPath l'avaluaria intentant aconseguir tots els camins que quadrin amb l'expressió. Podem veure això en el següent arbre:

Avaluació de l'expressió: `/classe/alumnes/alumne/nom`



El resultat, per tant, serien els dos resultats possibles següents: `<nom>Francisca</nom>` i `<nom>Francesc</nom>`

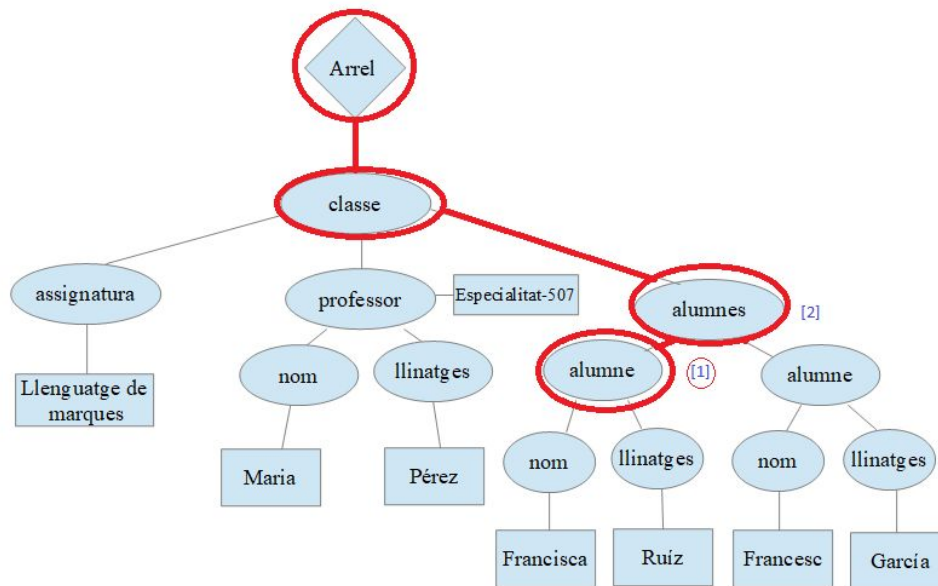
7.3 Navegació per l'arbre: Expressions

Si se sap que una expressió retornarà diversos resultats, però només se'n vol un d'específic, es pot fer servir un número envoltat per claudàtors `[]` per indicar quin és el que es vol aconseguir.

Exemple: Per retornar només el primer alumne es pot fer el següent: `/classe/alumnes/alumne[1]` dels dos nodes disponibles com a fills de `<alumnes>`, només se seleccionarà el primer:

O sigui:

```
<alumne>  
  <nom>Francisca</nom>  
  <llinatges>Ruíz</llinatges>  
</alumne>
```



7.3 Navegació per l'arbre: Expressions

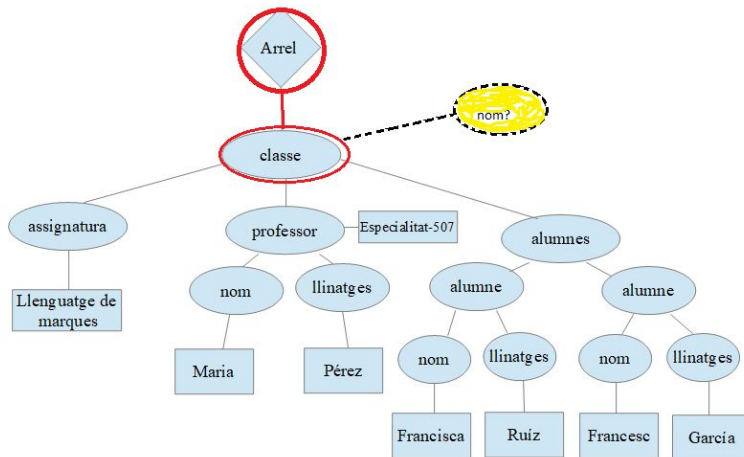
Es poden fer servir els claudàtors en qualsevol lloc de l'expressió per fer determinar quina de les branques es triarà.

Exemple: es pot obtenir només el nom del segon alumne amb una expressió com aquesta:
`/classe/alumnes/alumne[2]/nom`

Sempre s'ha d'anar en compte en escriure les expressions XPath, ja que si el camí especificat no es correspon amb un camí real dins de l'arbre, no es retornarà cap resultat.

Per exemple: `/classe/nom`

Com que en arribar al node "classe" no se'n trobarà cap anomenat <nom>, no retornarà cap resultat, com es pot veure a l'arbre:



7.3 Navegació per l'arbre: Expressions

Exemple: Si tenim el document XML

```
<?xml version="1.0" encoding="UTF-8"?>
<cine>
  <pelicula categoria="acció">
    <titol idioma="anglès">Mad Max</titol>
    <director>George Miller</director>
    <tiquet>8</tiquet>
    <repartiment>Tom Hardy</repartiment>
    <repartiment>Charlize Theron</repartiment>
    <repartiment>Nicholas Hoult</repartiment>
  </pelicula>
  <pelicula categoria="animació">
    <titol idioma="anglès">Inside Out</titol>
    <director>Pete Docter</director>
    <tiquet>10</tiquet>
    <repartiment>Amy Poehler</repartiment>
    <repartiment>Phyllis Smith</repartiment>
    <repartiment>Bill Hader</repartiment>
  </pelicula>
</cine>
```

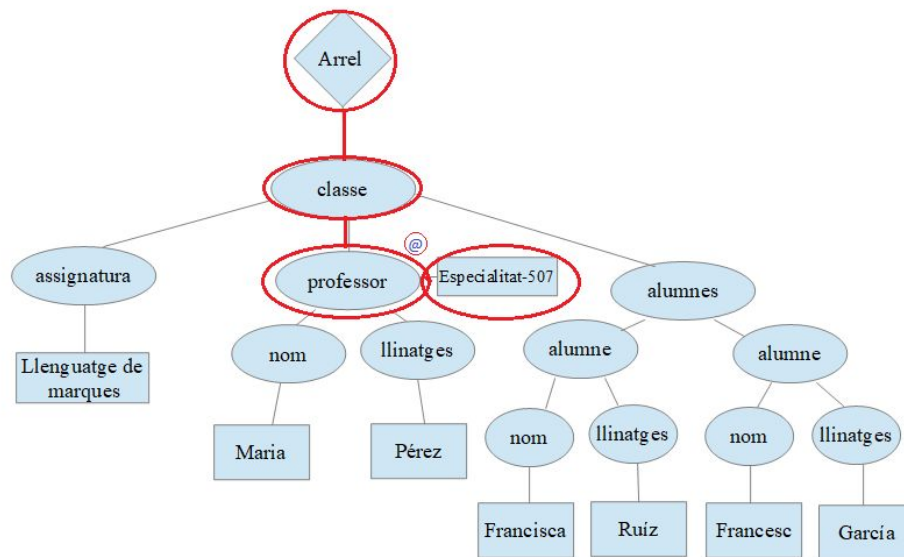
Podriem emprar les següents expressions XPath, per obtenir els resultats següents:

Expressió XPath	Resultat
/cine/pel·lícula[1]	Selecciona el primer element pel·lícula de l'element cine
/cine/pel·lícula[last()]	Selecciona el darrer element pel·lícula de l'element cine
/cine/pel·lícula[last()-1]	Selecciona el penúltim element pel·lícula de l'element cine
/cine/pel·lícula[position()<3]	Selecciona els dos primers elements pel·lícula de l'element cine
//titol[@idioma]	Selecciona tots els elements títol que ténen l'atribut idioma
//titol[@idioma='anglès']	Selecciona tots els elements títol que ténen l'atribut idioma amb el valor anglès
/cine/pel·lícula[tiquet>8]	Selecciona tots els elements pel·lícula de l'element cine que ténen un element tiquet major que 8
/cine/pel·lícula[tiquet>8]/titol	Selecciona tots els elements títols dels elements pel·lícula que ténen un element tiquet major que 8

7.4 Atributs

Obtenir els atributs d'un element

El valor dels atributs es poden aconseguir especificant el símbol @ davant el nom una vegada s'hagi arribat a l'element que el conté: /classe/professor/@especialitat



S'ha de tenir en compte que a diferència del que passa amb els elements, en obtenir un atribut no tindrem un element, sinó només el seu valor:

7.5 Comodins

Comodins

De la mateixa manera que en els sistemes operatius, es poden fer servir diversos comodins en les expressions XPath. Es poden veure els comodins a la següent taula:

Comodí	Significat
*	L'asterisc es fa servir per indicar tots els elements d'un determinat nivell.
.	Com en els directoris dels sistemes operatius el punt serveix per indicar el node actual.
..	Es fa servir per indicar el pare del node en el qual estem.
//	Les dobles barres indiquen que quadrarà amb qualsevol cosa des del node en el qual estem. Pot ser un sol element o un arbre de nodes.

7.5 Comodins

- **Amb l'asterisc:** Es poden obtenir tots els elements d'un determinat nivell.

Exemple: Amb aquesta expressió es poden obtenir tots els elements de dins del node professor

`/classe/professor/*`

Aquesta expressió retornarà per separat els dos nodes fills de <professor> (<nom> i <l·linatges>):

`<nom>Maria</nom>`

`<l·linatges>Pérez</l·linatges>`

7.5 Comodins

- Podem fer servir les dobles barres (//):

Exemple: Per obtenir tots els elements `<nom>` del fitxer independentment del lloc on siguin dins del document XML

```
//nom
```

El resultat serà:

```
<nom>Maria</nom>
```

```
<nom>Francisca</nom>
```

```
<nom>Francesc</nom>
```


7.5 Comodins

Exemple: Posar les dobles barres en qualsevol lloc dins de l'expressió per indicar que hi pot haver qualsevol cosa enmig a partir del lloc on apareguin

`/classe/alumnes//nom`

Donarà els dos noms dels alumnes:

`<nom>Francisca</nom>`

`<nom>Francesc</nom>`

Tot i que facilita la creació d'expressions no és gaire recomanable abusar del comodí "/" per motius d'eficiència. Les expressions amb aquest comodí requeriran molts més càlculs per ser avaluades, i per tant les expressions tardaran més a donar resultats.

7.6 Eixos XPath

Eixos XPath

Per avaluar les expressions XPath s'explora un arbre, de manera que també es proporcionen una sèrie d'elements per fer referència a parts de l'arbre. Aquests elements s'anomenen eixos XPath:

Eix	Significat
self::	El node en el qual està el procés (fa el mateix que el punt)
child::	Fill de l'element actual
parent::	El pare de l'element actual (idèntic a fer servir..)
Attribut::	Es fa servir per obtenir un atribut de l'element actual (@)

Alguns d'aquests eixos pràcticament no es fan servir perquè generalment és més còmode i curt definir les expressions com hem fet fins ara.

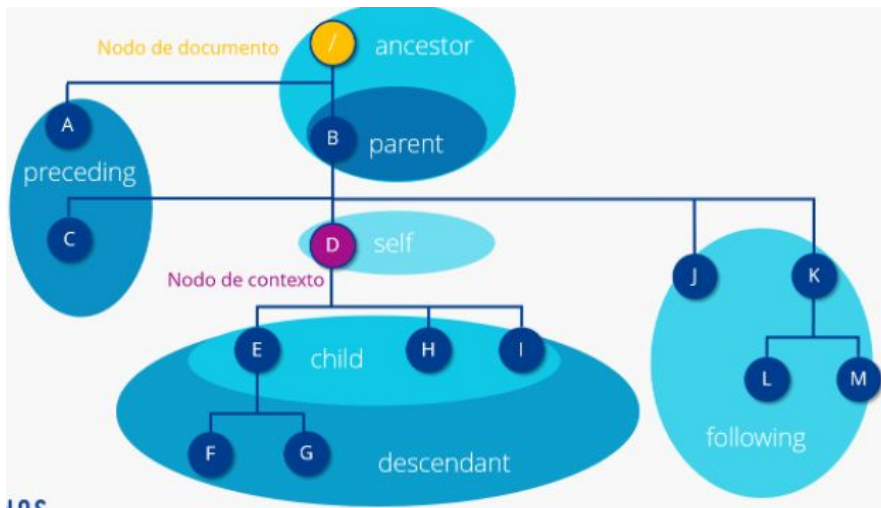
És més còmode fer servir una expressió com aquesta: /classe/professor/nom

I no fer servir la versió equivalent fent servir els eixos: /child::classe/child::professor/child::nom

7.6 Eixos XPath

Hi ha més eixos XPath:

Eix	Significat
descendant::	Tots els descendents del node actual
descendant-or-self::	El node actual i els seus descendents
ancestor::	Els ascendents del node
ancestor-or-self::	El node actual i els seus ascendents
preceding::	Tots els elements precedents al node actual
preceding-sibling::	Tots els germans precedents
following::	Elements que segueixen el node actual
following-sibling::	Germans posteriors al node actual
namespace::	Conté l'espai de noms del node actual



7.7 Condicions

Condicions

Un apartat interessant de les expressions XPath és poder afegir condicions per a la selecció de nodes. A qualsevol expressió XPath se li poden afegir condicions per obtenir només els nodes que compleixin la condició especificada.

- La selecció de nodes es fa especificant un predicat XPath dins de claudàtors [].

Exemple: Aquesta expressió selecciona només els professors que tenguin un element <nom> com a fill de <professor>.

```
/classe/professor[nom]
```

Si s'aplica l'expressió a l'exemple que hem fet servir per fer la vista d'arbre, el resultat serà el node <professor> que té dins seu <nom>:

```
<professor especialitat="507">  
  <nom>Maria</nom>  
  <l·linatges>Pérez</l·linatges>  
</professor>
```

7.7 Condicions

- Normalment la complexitat de les condicions va més enllà de comprovar si el node existeix, i es fan servir per comprovar si un node té un valor determinat.

Exemple: Per exemple, per obtenir els professors amb el nom “Maria”.

```
/classe/professor[nom="Maria"]
```

- Les condicions es poden posar en qualsevol lloc del camí i n'hi pot haver tantes com fasi falta.

Exemple: Obtenir el llinatge del professor amb el nom "Maria".

```
/classe/professor[nom="Maria"]/llinatges
```

Que donarà de resultat:

```
<llinatges>Pérez</llinatges>
```

7.7 Condicions

- De la mateixa manera que per obtenir-ne els valors, es poden fer comparacions amb els valors dels atributs especificant el seu nom darrera el símbol "@".

Exemple: Per saber si un element té l'atribut "especialitat".

```
/classe/professor[@especialitat]
```

Retornarà:

```
<professor especialitat="507">  
  <nom>Maria</nom>  
  <l·linatges>Pérez</l·linatges>  
</professor>
```

- De la mateixa manera que amb els elements, es poden posar condicions als atributs per saber si el seu valor té un determinat valor, etc.:

Exemples:

```
/classe/professor[@especialitat="507"]
```

```
/classe/professor[@especialitat>=507]
```

7.7 Condicions

- Es poden mesclar les expressions amb condicions sobre atributs i sobre elements per aconseguir expressions més complexes especificant-les una al costat de l'altra.

Exemple: Obtenir el professor que té l'atribut especialitat a "507" i que nom "Maria".

```
/classe/professor[@especialitat="507"][nom="Maria"]
```

- La funció *not()* es fa servir per negar les condicions.

Exemple: /classe/professor[not(@especialitat)]

- Les expressions poden ser tan complexes com es vulgui.

Exemple: Obtenir la llista dels llinatges dels alumnes del professor de tipus "507" que nom "Maria".

```
/classe/professor[@Especialitat="507"][nom="Maria"]/../alumnes/alumne/linatges
```

Que retornarà els dos llinatges:

```
<linatges>Ruíz</linatges>  
<linatges>García</linatges>
```

7.8 Funcions

Funcions

XPath ofereix una gran quantitat de funcions destinades a manipular els resultats obtinguts. Les funcions que ofereix XPath es classifiquen en diferents grups:

- Per manipular conjunts de nodes: es pot obtenir el nom dels nodes, treballar amb les posicions, comptar-los, etc.
- Per treballar amb cadenes de caràcters: permeten extreure caràcters, concatenar, comparar... les cadenes de caràcters.
- Per fer operacions numèriques: es poden convertir els resultats a valors numèrics, comptar els nodes, fer-hi operacions, etc.
- Funcions booleanes: permeten fer operacions booleanes amb els nodes.
- Funcions per dates: permeten fer operacions diverses amb dates i hores.

Nota: Pots consultar com aplicar les funcions consultar l'especificació XPath (<http://www.w3.org/TR/xpath-functions>).

7.8 Funcions

Funcions

Podem destacar algunes funcions a la següent taula:

Funció	Ús
name()	Retorna el nom del node
sum()	Retorna la suma d'una seqüència de nodes o de valors
count()	Retorna el nombre de nodes que hi ha en una seqüència
avg()	Fa la mitjana dels valors de la seqüència
max()	Retorna el valor màxim de la seqüència
min()	Dóna el valor mínim de la seqüència
position()	Diu en quina posició es troba el node actual
last()	Retorna si el node actual és l'últim
distinct-values()	Retorna els elements de la seqüència sense duplicats
concat()	Uneix dues cadenes de caràcters
starts-with()	Retorna si la cadena comença amb els caràcters marcats
contains()	Ens diu si el resultat conté el valor
string-length()	Retorna la llargada de la cadena
substring()	Permet extreure una subcadena del resultat
string-join()	Uneix la seqüència amb el separador especificat
current-date()	Ens retornarà l'hora actual
not()	Inverteix el valors booleans

7.8 Funcions

Exemple: Amb les funcions es podran fer peticions que retornin valors numèrics. Per exemple, "quants alumnes tenim?".

```
count(/classe/alumnes/alumne)
```

Que retornarà el nombre d'alumnes del fitxer: 2

Exemple: O bé retornar cadenes de caràcters. Per exemple, unir tots els noms separant-los per una coma.

```
string-join(/nom,",")
```

Exemple: Que retornarà en un únic resultat els noms separats per una coma.

```
Maria,Francisca,Francesc
```

Exemple: També es poden posar les funcions en els predicats. Per exemple, aquesta expressió ens retornarà els alumnes amb llinatge que comenci per "r".

```
/classe/alumnes/alumne[starts-with(linatges,"r")]
```

Exemple: En aquesta expressió volem obtenir el segon alumne de la llista.

```
/classe/alumnes/alumne[position()=2]
```