

Treball Fi de Carrera

**SINDICACIÓ DE
CONTINGUTS WEB**

Estudiant: José Luís Luján Gistau

Titulació: Enginyeria Tècnica Informàtica de Gestió

Consultor: Oscar Celma Herrada

Data: 19 de Juny de 2006

AGRAÏMENTS

En especial al suport de la meva família i amics i en concret a Dolors L, Carles L, Rosa P, Estela S, Carles M, Monica Q, Puri R per la seva ajuda desinteressada.

“Si vis pacem, para Bellum”

RESUM DE TFC

Aquest treball és el resultat del procés d'aprenentatge que he realitzat durant el període comprès des de Febrer de 2006 fins maig de 2006 i centra el seu estudi en el llenguatge XML i en el seu ús com a mitjà per compartir informació entre sistemes i aplicacions diferents gràcies als formats RSS / Atom.

Partint de la creació de documents en format RSS / Atom ens mostra la seva utilitat com a contenidors de dades i s'endinsa tant en les transformacions d'aquests documents en d'altres seguint els patrons de transformació creats segons les fulles d'estil XSLT, com en les cerques de parts dels documents que es poden fer amb el llenguatge de consulta XQuery amb l'ajut del llenguatge XPath per localitzar les dades interessants.

Un altre punt important d'aquest treball és l'estudi del sistema gestor de bases de dades que utilitzen els documents XML com a magatzems de dades de la mateixa manera que els sistemes, i com fer consultes des d'una aplicació web mitjançant Apache i PHP per tal de poder accedir a la BD amb la consulta en XQuery.

I finalment, per mostrar la utilitat de totes aquestes eines treballant conjuntament, s'ha desenvolupat una aplicació que permet realitzar cerques en documents RSS via web utilitzant com a SGBD la base de dades eXist.

INDEX

AGRAÏMENTS	2
INDEX DE FIGURES.....	5
CAPÍTOL 1. INTRODUCCIÓ	7
1.1 CONTEXT GENERAL	7
1.2 MOTIVACIÓ.....	7
1.3 OBJECTIUS DEL TFC.....	8
1.3.1 OBJECTIUS GENERALS	8
1.3.2 OBJECTIUS ESPECÍFICS	8
CAPÍTOL 2. SINDICACIÓ DE CONTINGUTS A LA WEB.....	9
2.1 INTRODUCCIÓ	9
2.2 HISTÒRIA DE LA SINDICACIÓ	10
CAPÍTOL 3. LA TECNOLOGIA XML	12
3.1 SINTAXI: XML BEN FORMAT	13
3.2 DECLARACIÓ D'UN DOCUMENT XML.....	13
3.3 ETIQUETES I ATRIBUTS.....	14
3.4 COMENTARIS.....	15
CAPÍTOL 4. FORMATS RSS I ATOM	16
4.1 FORMAT RSS	16
4.2 FORMAT ATOM	18
4.3 ALTRES ETIQUETES DE ATOM	20
4.4 COMPARATIVA RSS 2.0 I ATOM 1.0.....	21
CAPÍTOL 5. INTRODUCCIÓ A LES BD:XML NATIVES	22
5.1. ASPECTES GENERALS DE LES BD:XML	22
5.1.1 INTRODUCCIÓ.....	22
5.1.2 CARACTERÍSTIQUES COMUNES.....	22
CAPÍTOL 6. eXist: EXEMPLE DE BD:XML NATIVA	24
6.1 INDEXAT I EMMAGATZEMATGE	24
6.1.1 ESQUEMA DE NUMERACIÓ D'EXIST	25
6.1.2 ÍNDEXS I ORGANITZACIÓ DE LES DADES	25
6.2 MODES DE TREBALL AMB EXIST	26
CAPÍTOL 7. EL LLENGUATGE DE CONSULTA XQuery	28
7.1 DEFINICIÓ DE XQuery.	28
7.2 REQUERIMENTS TÈCNICS DE XQuery.....	29
7.3 CONSULTES EN XQuery.....	30
7.3.1 REGLES GENERALS.....	32
7.4 FUNCIONS D'ENTRADA	33
7.5 EXPRESSIONS CONDICIONALS.....	33
7.6 QUANTIFICADORS EXISTENCIALS.....	33
7.7 OPERADORS I FUNCIONS PRINCIPALS.....	34
7.8 COMENTARIS.....	35
CAPÍTOL 8. EXEMPLES DE CONSULTES XQuery A eXist	36
PRIMER EXEMPLE DE CONSULTA	37
SEGON EXEMPLE DE CONSULTA	37
TERCER EXEMPLE DE CONSULTA	38

QUART EXEMPLE DE CONSULTA.....	38
CINQUÈ EXEMPLE DE CONSULTA	38
SISÈ EXEMPLE DE CONSULTA	39
CAPÍTOL 9. CAS PRÀCTIC: APLICACIÓ WEB PER CONSULTAR DOCUMENTS	
RSS	40
9.1 COMPONENTS DE L'APLICACIÓ	40
9.2 EXEMPLES DE CONSULTA.....	44
CAPÍTOL 10. CONCLUSIONS	49
CAPITOL 11.GLOSSARI	50
CAPÍTOL 12. BIBLIOGRAFIA	50
ANNEX A	52
A.1 INSTAL·LACIÓ JAVA	52
A.1.1 PROCÉS D'INSTAL·LACIÓ.....	52
A.1.2 COMPROVACIONS.	53
ANNEX B.....	53
B.1 INSTAL·LACIÓ D'EXIST	53
B.1.1 COMPROVAR SI EL SERVIDOR ESTÀ ENEGAT	54
B.1.2 ARXIUS INSTAL·LATS.....	54
ANNEX C: Document RSS de la BD del cas pràctic	55
C.1 "rss0.XML"	55
ANNEX D: INSTAL·LACIÓ DE APACHE EN WINDOWS	58
D.1DESCÀRREGA DE APACHE 2.0.50	58
D.1 INSTAL·LACIÓ.....	58
D.3 COMPROVACIÓ DEL FUNCIONAMENT.....	59
ANNEX E: INSTAL·LACIÓ DE PHP WINDOWS	59
E.1 INSTAL·LACIÓ DE PHP 5.0.1.....	59
E.2 CONFIGURACIÓ DEL ARXIU PHP.INI	60
E.3 httpd.conf PER APACHE 2.0.X.....	61
INDEX DE FIGURES	
Figura 1.Sindicació de continguts.....	9
Figura 2. Icones relacionats amb sindicació.....	10
Figura 3.Ordre d'execució i resultat de les cinc clàusules possibles.....	32
Figura 4. eXist Database Login	36
Figura 5. eXist Admin Client	37
Figura 6. Pantalla de presentació de la aplicació Web.....	41
Figura 7. Formulari de Recerca i mostra de resultats.....	42
Figura 8. Opcions de recerca	42
Figura 9. Número màxim de recerques	43
Figura 10.Caixa de text de recerca.....	43
Figura 11. Opcions wrap i indent.....	43
Figura 12. Botons Cerca i Neteja de Formulari	43
Figura 13. IFrame de mostra de resultats.....	44
Figura 14. Consulta Mostrar tot per autor.....	45
Figura 15. Consulta Mostrar tot per títol	45

Figura 16. Consulta Mostrar tots els autors.....	46
Figura 17. Consulta Mostrar tots els títols	47
Figura 18. Consulta Cerca per autors.....	47
Figura 19. Consulta Cercar per títols.....	48
Figura 20. Apache funcionant.....	59

INDEX DE TAULES

Taula 1. Versions RSS i Atom	16
Taula 2. Opcions de consulta	28
Taula 3. Possibles clàusules en una consulta XQuery.....	31
Taula 4. Operadors i funcions en XQuery	34

CAPÍTOL 1. INTRODUCCIÓ

1.1 CONTEXT GENERAL

El present treball final de carrera (TFC) es basa en la presentació del concepte de sindicació de continguts a la Web i els mecanismes i tecnologia que actualment s'està utilitzant per aconseguir-ho.

La sindicació es basa en uns formats anomenats [RSS / Atom](#) els quals a la vegada estan basats en un llenguatge de representació de la informació, anomenat XML. Aquest llenguatge està àmpliament adoptat com a format robust i flexible que facilita la portabilitat de les dades entre sistemes. Aquestes dades es poden desar i recuperar mitjançant els diferents sistemes de bases de dades XML natives (BD:XML). Aquestes BD:XML proporcionen una sèrie de llenguatges per accedir a les dades contingudes en un document XML com per exemple; [XQuery](#) i [XPath](#).

1.2 MOTIVACIÓ

[L'XML](#) (*eXtensible Markup Language*) va ser definit el 1998 pel [W3C](#) (*World WideWeb Consortium*) per a representar informació de manera independent a qualsevol format. Per tal d'accedir a aquesta informació, d'extreure-la i de manipular-la, el W3C està treballant en la definició d'un llenguatge de consultes per a XML anomenat *XQuery*.

L'emmagatzemament i el tractament d'aquesta informació es pot gestionar des d'algunes bases de dades relacionals convencionals. Tanmateix, la solució òptima consisteix en gestionar aquestes dades en el seu format nadiu, sense haver de recórrer a cap conversió de formats, mantenint intacta l'estructura dels documents.

Per mitjà de l'estudi previ d'aquestes tecnologies, del plantejament i de la resolució d'un cas pràctic, el present treball de final de carrera (*TFC*) pretén demostrar la aplicació d'aquesta tecnologia al concepte de sindicació en la Web basant integrament en llenguatge XML sota els formats RSS/Atom.

1.3 OBJECTIUS DEL TFC

1.3.1 OBJECTIUS GENERALS

Els objectius generals del TFC consisteixen en:

- Comprendre el concepte de sindicació a la Web.
- Conèixer els diferents formats de sindicació actuals RSS i Atom.
- Assolir els coneixements necessaris d'*XML*.
- Comprendre un sistema de bases de dades nadiu (*eXist*).
- Conèixer diferents llenguatges de consultes per a treballar amb *XML* (*XQuery* *XPath*).
- Realitzar un cas pràctic per a consolidar tots els coneixements assolits fent consultes a una base de dades *eXist* de feeds mitjançant una pàgina Web.

1.3.2 OBJECTIUS ESPECÍFICS

Per tal d'assolir els objectius generals ha calgut realitzar una sèrie d'objectius específics (entesos com a tasques), els quals han estat agrupats en Proves d'avaluació Continuada (*PAC*), seguint el model pedagògic de la Universitat Oberta de Catalunya (*UOC*):

- *PAC1*: Pla de Treball.
- *PAC2*:
 - Estudi del concepte de Sindicació a la Web.
 - Estudis dels principals formats de sindicació RSS i Atom.
 - Estudi d'*XML*.
 - Estudi de les *BD:XML* natives.
 - Instal·lació *eXist*.
- *PAC3*:
 - Estudi d'*XQuery 1.0*
 - Estudi *XPath*.
 - Estudi *HTML*.
 - Estudi *PHP 5*
 - Plantejament i resolucions del cas pràctic.
 - Lliurament final:
 - Memòria.
 - Presentació virtual.

CAPÍTOL 2. SINDICACIÓ DE CONTINGUTS A LA WEB

2.1 INTRODUCCIÓ

La sindicació de continguts web, també anomenat “redifusió” és la forma que tenen alguns llocs web de posar a disposició d’altres llocs subscriptors els continguts als quals els usuaris accedeixen freqüentment sense necessitat d’entrar en el propi lloc web. És com enviar un correu electrònic a cada usuari que així ho volgués, no només per a avisar-li que hi ha un contingut nou, si no, a més oferir-li aquest contingut (de forma total o parcial) al costat d’altres continguts d’altres llocs web.

El contingut es distribueix mitjançant uns **canals de sindicació**, aquests canals no són més que una selecció de continguts als que l’usuari pot subscriure’s els quals tenen una adreça molt semblant a la de qualsevol pàgina web. Els usuaris poden llegir aquests canals amb un programari denominat **agregador** (en anglès newsreader o feed reader), els quals permeten rebre la informació a mesura que es va carregant en el lloc web origen.

La sindicació de continguts es realitza mitjançant arxius XML dotats d’una estructura idònia per a presentar resums de contingut ja que permet publicar metadades dels continguts d’un lloc web utilitzant els formats estàndard de sindicació RSS i Àtom. Aquests formats permeten ser processats per aplicacions estàndard (parseado).

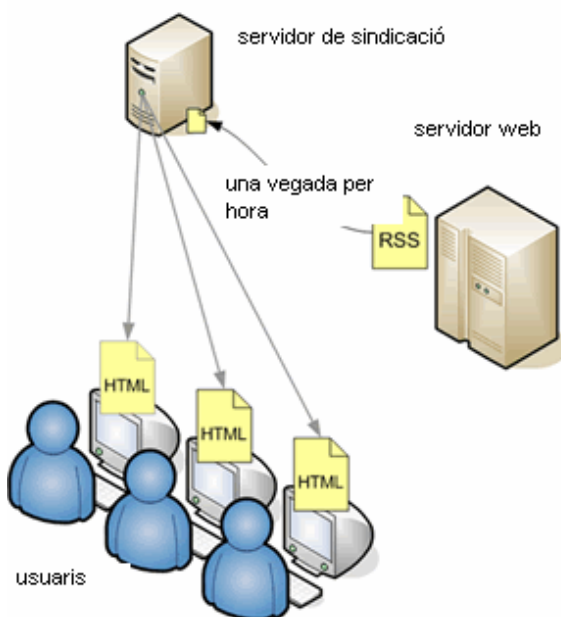


Figura 1. Sindicació de continguts

La figura 1 mostra com s'enllaçaria, en una sindicació de continguts, els usuaris amb els llocs web.

Els drets de sindicació de continguts web solen ser gratuïts i no sol haver un contracte entre les dues parts, sinó una llicència de normes d'ús.

2.2 HISTÒRIA DE LA SINDICACIÓ

Encara que la sindicació s'ha començat a utilitzar de forma intensa en els últims anys, no és una tecnologia nova. El format més utilitzat i conegut és el RSS que va ser inventat per Netscape l'any 1999 com una forma de donar a conèixer els seus propis canals, però no va tenir molt èxit perquè es limitava a un selecte grup de llocs comercials.

Una vegada que AOL va comprar la signatura, RSS va anar quedant en l'oblit fins que Dave Winer i la seva empresa Userland van rescatar-lo amb la finalitat de millorar-lo.

Actualment, navegadors com ara Firefox o Opera estan incorporant utilitats de sindicació mitjançant marcadors dinàmics, que proporcionen un accés immediat a gran quantitat de continguts en el mateix moment que estan sent publicats. Aquests continguts són de la elecció del propi usuari.

La potència de la sindicació radica en la seva simplicitat i s'aconsegueix que es pugui accedir a la informació d'Internet de manera fiable i barata.

Les pàgines sindicades solen tenir icones com les que es mostren a la figura 2, els quals tenen un enllaç que ens duu a una altra pàgina on tenim el codi XML necessari per a construir el format RSS.



Figura 2. Icones relacionats amb sindicació

Actualment, els usos de la sindicació de continguts s'utilitza per a syndicar temes com ara:

- Butlletins de notícies.
- Publicacions de nou material disponible, com per exemple nous llibres, CD's de música etc.
- **Podcasting** que consisteix a crear arxius de so en mp3 o ogg i syndicar-los de manera que permeti subscriure's i descarregar-los perquè l'usuari ho escolti en el moment que vulgui.
- **Bitàcores** (*weblog o blog*) que és un lloc web que disposa de comentaris de caràcter personal, ordenats cronològicament. Aquestes anotacions permeten, en general, ser enriquides pels lectors del *site*, mitjançant comentaris. L'escriptor de bitàcores (*weblogger o blogger*), sol visitar alhora altres bitàcores i disposa en el seu espai d'un llistat amb aquells que li són útils o afins, conformant grups o petites comunitats que interactuen entre si etc.

CAPÍTOL 3. LA TECNOLOGIA XML

Aquest apartat resumeix el llenguatge XML, sintetitzant els coneixements necessaris per tal de poder treballar de manera bàsica amb documents XML. És pretén fer una presentació de XML, tot presentant un resum de la sintaxi, tal que permeti donar una idea general de les tecnologies associades a XML per tal de definir i generar documents descriptius de recursos.

Extensible Markup Language (XML) és fonamental a l'hora d'implementar la tecnologia del present TFC. La majoria dels feed (RSS i Atom) estan implementats amb XML. Moltes de les aplicacions que utilitzen els feed, utilitzen les tecnologies lligades amb XML per a poder processar, analitzar i, finalment, poder mostrar a l'usuari els feed d'una manera convenient.

XML és un llenguatge d'etiquetes orientat a que una màquina pugui llegir les dades que poden tenir qualsevol tipus de informació, les quals són perfectament llegibles per l'ésser humà. Un programa d'anàlisi de XML s'encarregarà de processar aquests documents escrits en XML.

XML té dos metallenguatges amb els que definir els llenguatges que ens permetran escriure documents XML: el DTD i el XML Schema.

XML s'utilitza de manera semblant a HTML, és a dir, mitjançant etiquetes, es diu que és un metallenguatge que descriu la sintaxi de com han de ser escrites les dades. Algunes diferències que es poden trobar respecte a l'HTML és que és un llenguatge purament estructural (no inclou cap referència al disseny), llavors ens farà falta un llenguatge de disseny. No obstant això, XML presenta unes característiques prou importants que el permeten actuar com llenguatge d'interoperabilitat entre sistemes com són:

- Permet definir etiquetes pròpies.
- Se li poden assignar atributs a les etiquetes.
- Les etiquetes i els atributs es defineixen mitjançant un esquema (DTD o XML Schema).
- L'estructura i el disseny estan desacoblats.

3.1 SINTAXI: XML BEN FORMAT

Un document XML es diu que està ben format si compleix totes les regles de SINTAXI. Els documents XML tenen una estructura jeràrquica molt estricta i els documents ben formats l'han de complir. Si a més a més de ser ben formats segueixen una estructura i semàntica determinada per un DTD o XML Schema, es diu que el document és vàlid.

Les regles per les que un document es considera ben format són les següents:

- Tot document XML té un únic element arrel.
- Les etiquetes han d'estar correctament niades.
- Els noms dels elements han de complir certes convencions de nomenclatura.
- XML mantindrà els espais en blanc i els salts de línia dins de les etiquetes.
- XML fa distinció entre majúscules i minúscules.
- XML té un conjunt de caràcters reservats (<, >, &, ", '), els quals, si els volem utilitzar ho haurem de fer usant les entitats següents. <, >, &, ", '.

Els documents XML només contenen text amb caràcters ASCII. El text inclòs dintre dels caràcters "<" menor que i ">" major que , són les parts del document que entenen els processadors de XML.

3.2 DECLARACIÓ D'UN DOCUMENT XML

La declaració de XML és opcional en l'especificació XML 1.0, hauria d'estar present. En el cas d'estar present, haurà de ser la primera etiqueta del document i no li hauria de precedir ningun espai o caràcter.

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

Els atributs que acompanya al element XML són els següents.

- **version:** ens indica la versió d'XML que s'usarà en el document.
 - **encoding:** és un camp opcional que indica la codificació que s'usarà en el document (per defecte es UTF-8).
 - **standalone:** és un camp opcional que ens indica si el document va acompanyat d'un DTD o no.
-

3.3 ETIQUETES I ATRIBUTS

El text marcat amb “<...>” es coneix com etiqueta. L'etiqueta està formada per “<”, un nom element identificador i “>”. Hi ha una etiqueta inicial i la final. Totes dues porten el mateix nom de l'element, i enmig de les dues etiquetes el contingut que pot ser ASCII o altres etiquetes. A vegades no hi ha contingut, es diu que són les etiquetes buides.

Les etiquetes inicials comencen en un signe “<” i acaben en “>” mentre que les etiquetes finals comencen per “</” i acaben per “>”. Els identificadors es poden concretar amb atributs mitjançant la parella atribut = “valor”. A més, els atributs no es poden repetir dintre de una etiqueta.

```
<element atribut = "valor"> text </element>
```

Tots els documents d' XML tenen almenys un element que s'anomena element del document o element arrel. Aquesta arrel s'obre al començar el document i es tanca al final. Tota etiqueta que s'obri cal que sigui tancada respecte el seu respectiu ordre. Cal que les etiquetes estiguin correctament niades. En *RSS0.91* el document de l'element és el següent:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
  <rss version="0.91">
    <channel>
      <title>L'art de la guerra</title>
      <link>www.amazon.com</link>
      <description>Filosofia i principis de la
guerra</description>
      <language>es</language>
    </channel>
  </rss>
```

El codi anterior mostra un document ben format on a la seva primera línia hi ha la declaració i després una etiqueta rss (element arrel) que conté una sèrie d'elements (channel, title, link, description, language).

Podem observar que tots els elements oberts estan tancats i correctament niats. La capacitat de elements niats és clau per a l'èxit d'un llenguatge de marques. Un

element aïllat pot proporcionar només una part de la descripció d'un bloc de continguts, mentre que els elements niats afegixen moltes possibilitats.

Però la relació entre elements no només són avantatges, si no també pot originar molts problemes. Un disseny acurat dels models de contingut pot garantir que els creadors de contingut afegiran els elements necessaris, i s'evitarà la creació de combinacions problemàtiques.

3.4 COMENTARIS

Per tal de facilitar la lectura d'un document XML s'utilitzen els comentaris. Els comentaris comencen pel caràcter `<!--` i acaben amb els `-->`. Tal com es mostra un exemple.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rss version="0.91">
<!--aquí comença la declaració del canal RSS -->
  <channel>
    <link>
      http://www.amazon.com/exec/obidos/tg/browse/-/229535/1/?rank=%2Bsalesrank%26dev-t=amznRss
    </link>
    <description>Up-to-date listings of Amazon.com's top items</description>
    <language>en-us</language>
    <docs>http://www.amazon.com/exec/obidos/subst/xs/syndicate.html</docs>
  </channel>
</rss>
```

CAPÍTOL 4. FORMATS RSS I ATOM

Els formats RSS (Really Simple Syndication) i Atom estan basats en el llenguatge XML i són els utilitzats per a syndicar continguts web. Hi ha hagut diverses versions, moltes de les quals fruit de la competència entre les companyies líders d'aquests formats. Seguidament es presenta una taula resum amb l'evolució d'aquests formats ordenats cronològicament:

Tecnologia	Versió	Data	Companyia
RSS	0.9	Març de 1999	Netscape
RSS	0.91	Juny de 2000	UserLand
RSS	1.0	Desembre de 2000.	RSS-Dev Working Group
RSS	0.92	Desembre de 2000	UserLand
RSS	2.0	Setembre de 2002	UserLand
Atom	0.3	Desembre de 2003	Mark Nottingham
Atom	1.0	Juny de 2005	Atompub working group

Taula 1. Versions RSS i Atom

4.1 FORMAT RSS

El format *RSS2.0* (*Really Simple Syndication*) està desenvolupat per una altra companyia diferent a la que va desenvolupar *RSS1.0*, llavors no es pot dir que substitueix el format *RSS1.0* i ambdós formats poden conviure a la vegada en la xarxa.

El més basic d'un format RSS ho té la versió 0.91, de la qual s'extreuen les següents etiquetes que també estaran incloses en la *RSS2.0*.

```
<rss>
  <channel>
    <title>Amazon.com - Electronics: Accessories & Supplies:
    All Accessories & Supplies</title>
    <link>http://www.amazon.com/exec/obidos/tg/browse/-
    /281407/1/?rank=%2Bsalesrank%26dev-t=amznRss</link>
```

```

    <description>Up-to-date listings of Amazon.com's top
items (by Sales Rank).</description>
    <language> en-us</language>
    <image>
        <title>Amazon.com--Earth's Biggest
Selection</title>
        <width>120</width>
        <height>30</height>
    </image>
    <item>
        <title> Logitech Wireless Music System for iPod
</title>
        <link>
http://www.amazon.com/exec/obidos/ASIN/B000AXQFJC/ref=nosim
Rss/webservices-20?dev-t=amznRss</link>
        <description>Logitech Wireless Music System for
iPod -- $139.99</description>
    </item>
</channel>
</rss>

```

<rss> És una etiqueta única que conté tot el contingut del *feed*, en el seu interior es col·loca l'atribut *version* on tindrem la versió *rss* utilitzada. L'etiqueta cal tancar-la amb **</rss>**.

<channel> és una etiqueta que descriu el contingut del canal.

Les etiquetes **<title>**, **<link>** i **<description>** donen informació del *feed* en la seva totalitat.

L'etiqueta **<languages>** és opcional. Indica el codi del llenguatge del contingut del *feed*.

<item>: pot tenir diversos items, *RSS0.91* deixa un màxim de 15 *items* però la versió *RSS2.0*, ja no té aquest límit. Per cada etiqueta que posem, cal posar **<title>**, **<link>** i **<description>** amb la seva corresponent etiqueta de finalització **</item>**. Quan acabem

d'afegir tots els items cal tancar l'etiqueta `</channel>`, per finalitzar amb l'etiqueta `</rss>`

Altres elements opcionals de **<channel>** són:

<category> On es col·loca la categoria de la informació, i a més a més pot tenir diverses dintre d'un mateix canal.

<cloud> És l'element que conté atributs que permeten la subscripció a un feed i ha d'estar sempre actualitzat.

<copyright> És el copyright relacionat amb un feed.

<docs> Aquest element conté la URL que assenyalava l'especificació.

<generator> Conté informació del programari que ha generat el feed.

<image> Conté la informació (en format gif, jpeg o png) de la imatge que es visualitzarà amb el feed.

<language> Conté l'extensió del llenguatge emprat.

<managingEditor> Conté la direcció de correu electrònic per a contactar amb l'editor del contingut.

<pubDate> Conté la data de publicació del feed.

<skipDays>. Conté la informació indexant al agregador els dies de la setmana que un feed no s'espera que s'actualitzi.

<ttl> Conté informació del període de temps que ha d'esperar l'agregador per a buscar un nou contingut.

<webMaster> Conté informació per a contactar amb el realitzador del feed.

4.2 FORMAT ATOM

Atom està format pel llenguatge XML i no es correspon ni es basa en cap versió de RSS, però és un format molt similar a aquest i, sobretot, té el mateix objectiu: permetre la distribució de continguts i notícies de llocs web.

La versió més actual és la 1.0 de juny de 2005. Les millores que suposa respecte a RSS (en qualsevol de les seves versions) fan que el seu ús s'estengui ràpidament malgrat de ser una mica més complicat. Un document Atom pot contenir més informació (i més complexa) i és més consistent que un document RSS.

L'estructura general és la següent:

`<feed>`

```
<title>dive into mark</title>
<link>
<modified>2003-12-3T18:30:02Z </modified>
<author><name>Mark Pilgrim</name></author>
  <entry>
    <title> Atom 0.3 snapshot</title>
      <link>
        <id> tag:diveintomark.org,2003:3.2397</id>
        <issued>2003-12-13T08:29:29-04:00</issued>
        <modified>2003-12-3T18:30:02Z</modified>
      </entry>
</feed>
```

<author> Conté el nom de l'autor del feed. Un feed pot tenir diversos autors i per tant diversos elements author. Com a mínim n'ha de tenir un.

<link> identifica una pàgina web relativa al feed. El tipus de relació ve definit per l'atribut *rel*. Un feed està limitat per un element link. L'element *link* té un atribut requerit, *href*, i cinc atributs opcionals: *rel*, *type*, *hreflang*, *title* i *length*.

href conté una URI pel recurs especificat, (normalment una pàgina Web).

rel conté el tipus de relació del enllaç que pot ser una URI completa o un dels següents valors predefinits (el valor per defecte es *alternate*):

- **alternate:** especifica que el enllaç es una representació alternativa de la entrada o del feed, per exemple un *permalink* o una versió HTML de la entrada, o bé la pàgina principal de un *blog*.
- **enclosure:** Un recurs associat que és massa gran en tamany i que pot requerir un tractament especial, com per exemple una gravació d'àudio o de vídeo.
- **related:** Un document relacionat amb la entrada o el feed.
- **self:** El feed en si mateix.
- **via:** La font de informació que ha propiciat la entrada.

Type: indica el tipus de recurs.

Hreflang: Indica el llenguatge referenciat.

title indica un text informatiu en format llegible per persones sobre el enllaç, normalment s'utilitza com nombre per el enllaç.

content Conté l'enllaç cap al contingut de l'entrada. L'element *content* és obligatori si no hi ha un element *link* alternatiu i hauria de ser-hi si no hi ha un element *summary*.

```
<content>aquí hi ha el contingut d'una entrada</content>
```

summary Conté un petit resum de l'entrada. L'element *summary* hauria d'oferir-se si no hi ha un element *content* per l'entrada, o si el contingut de *content* no està especificat en línia, (per exemple mitjançant un atribut *src* que conté un link exterior cap al contingut).

```
<summary>Qualsevol text.</summary>
```

4.3 ALTRES ETIQUETES DE ATOM

<category> L'element **category** té un atribut obligatori, *term* i dos atributs opcionals, *scheme* i *label*.

- *term*: identifica la categoria
- *scheme*: identifica el esquema de categorització mitjançant una URI.
- *label*: dona a *category* un nom llegible per persones.

<content> Aquest element conté o bé un enllaç al contingut de l'entrada, o bé al contingut en si mateix. En la majoria del casos, l'atribut *type* té per valor: *text*, *html* o *xhtml*, el qual indica el format del text que conté.

Per altra banda, si l'atribut *src* està present, aquest indica una URI on es pot trobar en contingut.

Si l'atribut *type* acaba amb *+xml* o */xml*, llavors un document xml del tipus definit està ontingut en línia.

Si l'atribut *type* comença amb *text*, llavors significa que un document en línia sense marcar està contingut en l'element.

<person >

author i *contributor* descriuen una persona, corporació o entitat similar. Té un element obligatori, *name* i dos opcionals *uri* i *email*.

- *name*: especifica un nom per la persona, en format llegible per els usuaris.
- *uri*: Conté la pàgina principal de la persona (normalment la seva pàgina personal, blog o similar)
- *email*: conté la direcció de correu electrònic de la persona.

4.4 COMPARATIVA RSS 2.0 I ATOM 1.0**Especificacions:**

- A partir de la versió RSS 2.0 serà difícil introduir nous canvis significatius, en canvi ATOM 1.0 deixa oberta l'especificació per a publicar versions o revisions futures conservant les actuals.
- En el format RSS, les entrades noves es diuen <item> mentre que en Atom es diuen <entry>. Atom no utilitza l'etiqueta <channel>.
- Atom utilitza l'etiqueta <id> per a identificar cada entrada en lloc de <guid>.
- Atom utilitza els namespace per a identificar les etiquetes permissibles.
- En Atom tots els links han d'incloure els atributs *rel*, *type* i *href* amb els valors *alternate*, *text/html* i URL respectivament.
- Publicació de protocols: Atom implementa millor els protocols utilitzats en les bitàcores.
- Respecte el contingut requerit: RSS 2.0 requereix un títol del feed, un link i una descripció, i no és necessari que tots aquests camps estiguin presents en cada ítem, mentre que Atom 1.0 requereix un títol (encara que aquest camp estigui buit), un únic identificador i l'última data d'actualització

CAPÍTOL 5. INTRODUCCIÓ A LES BD:XML NATIVES

5.1. ASPECTES GENERALS DE LES BD:XML

5.1.1 INTRODUCCIÓ

Les bases de dades són una part fonamental de totes les organitzacions, doncs en elles s'emmagatzema informació que serà utilitzada per aquestes. A més, cal dir que XML és part del present i segurament del futur de l'administració i gestió de dades, ja que ha permès crear un format estàndard per a processar informació. En les darrers anys, XML està provocant l'aparició de noves tecnologies, entre elles l'aparició d'una nova generació de bases de dades, les quals ara per ara es troben en una fase d'investigació i desenvolupament, però en un futur poden ser una alternativa a les bases de dades relacionals. Són les anomenades "native XML database", les bases de dades XML natives (BD:XML).

5.1.2 CARACTERÍSTIQUES COMUNES

Tot i que cada sistema gestor de bases de dades té característiques diferents, en general totes les BD:XML tenen una sèrie de característiques comunes:

Emmagatzematge:

Les bases de dades natives XML emmagatzemen els documents creant models lògics. Aquests models es basen directament en XML o en alguna de les tecnologies relacionades com DOM (Document Object Model) o Infoset que defineix la propietat del URI base dels elements d'informació, l'última especificació Infoset suporta XML Base, amb fi de determinar el valor d'aquesta propietat. Aquests dos models inclouen diferents nivells d'agregació i complexitat així com un suport complet per la gestió de dades semiestructurades. Els models són automàticament mapejats per les bases de dades XML natives al mecanisme d'emmagatzematge corresponent. El mapeig usat per aquestes bases de dades garantirà la correcta utilització del model associat al document original.

Col·leccions:

Aquest tipus de bases de dades treballa sobre col·leccions de documents. Això és similar al concepte relacional d'una taula tot i que divergeix en el sentit que no requereix que la col·lecció tingui un esquema associat. Per tant, es pot emmagatzemar

qualsevol document XML en una col·lecció sense necessitat de cap esquema i tot i això es poden fer consultes a través de tots els documents de la mateixa col·lecció.

Aquesta propietat s'anomena esquema-independent.

Cerques:

Actualment, XPath és el llenguatge de cerques més estès implementat pel diferents sistemes de BD:XML. XPath, però, no va ser dissenyat com a llenguatge de consultes de bases de dades i té diverses limitacions. Algunes d'elles s'han solucionat ampliant el llenguatge per exemple per permetre consultes a través de col·leccions XML. No obstant les carències més importants com agrupar, ordenar, fer "cross document join" no s'han pogut solucionar i per fer-ho es sol recórrer al llenguatge XSLT o si la BD:XML implementa XQuery llavors permet fer el tipus de cerques explicades a la secció anterior..

Processament de les dades:

El processament de dades hauria de ser una cosa realment senzilla però la realitat és que no és així. Moltes bases de dades necessiten recuperar tot el document XML per actualitzar-lo amb qualsevol API i posteriorment tornar a emmagatzemar el document. Això és degut a que encara no existeix un llenguatge estàndard que permeti fer l'actualització de manera senzilla i ràpida. Actualment, s'està desenvolupant un llenguatge (XUpdate) per fer aquesta tasca però encara no és un estàndard i moltes BD:XML encara no el suporten.

En aquest treball presentarem eXist com una BD:XML per poder desenvolupar el Treball Final de Carrera.

CAPÍTOL 6. eXist: EXEMPLE DE BD:XML NATIVA

eXist és un sistema de BD:XML open source, desenvolupat íntegrament en Java. El sistema d'emmagatzematge es fa a través de col·leccions jeràrquiques. Usant una versió estesa de XPath els usuaris poden consultar una part de la col·lecció jeràrquica o tot els documents continguts a la base de dades. Un millorat esquema d'indexació proporciona una ràpida identificació de relacions estructurals entre nodes del tipus pare-fill, antepassat-descents o previous-next.

La base de dades es ideal tant per petites com grans col·leccions de documents XML que s'actualitzen de tant en tant. eXist proporciona extensions a XPath estàndard i permet fer cerques per paraula clau. Per programadors permet accés a través d'HTTP, XML-RPC, SOAP i WebDAV.

6.1 INDEXAT I EMMAGATZEMATGE

Els llenguatges de consulta XML com XQuery i XPath usen expressions per navegar a través de les estructures lògiques i jeràrquiques dels documents XML que es modelen com un arbre. Una expressió XPath localitza nodes dins un arbre per exemple `book//section/title` selecciona tots els elements title que són fills dels elements secció que tenen com a ancestre un element anomenat book.

D'acord amb XPath 2.0 el resultat d'una expressió de camí és una seqüència de nodes diferents en un document ordenat. La seqüència de nodes pot ser filtrada per expressions de predicat. Una expressió de predicat va tancada entre claudàtors. Aquells nodes que no compleixen el predicat són descartats de la seqüència per exemple per trobar aquelles seccions el títol de les quals es XQuery podríem usar la següent expressió:

```
book//section[contains(title, 'XQuery')].
```

La subexpressió de predicat especifica una selecció basada en valor mentre que la subexpressió `book//section` denota una selecció estructural. Les seleccions basades en valors poden ser especificades en noms d'element, noms d'atribut/valor o strings de

text continguts en un element. Les seleccions estructurals són basades en les relacions estructurals entre nodes tal com ancestor-descendant o parent-child

6.1.1 ESQUEMA DE NUMERACIÓ D'EXIST

El esquema de numeració el qual modela el document arbre com un arbre complet k-ary on k és igual al màxim nombre de nodes fill de l'element en el document. Un únic node identificador és assignat a cada node per travessar l'arbre en ordre.

Els identificadors únics generats per aquest esquema de numeració té algunes propietats importants: per un identificador donat pot ser fàcilment determinat a l'identificador del node del seu pare, germans o fills. Per obtenir-lo només cal aplicar la fórmula següent:

$$\text{Parent}_i = (i-2/k)+1$$

eXist proporciona una implementació completa del llenguatge de consulta XPath. Suporta tots els axis del llenguatge. Usant l'esquema de numeració de eXist podem fàcilment computar el identificador del node pare per cada node donat per avaluar l'expressió. També podem computar els identificadors dels nodes germans i fills. D'aquesta manera tots axis de la navegació poden ser implementats en el nostre esquema.

6.1.2 ÍNDEXS I ORGANITZACIÓ DE LES DADES

eXist usa quatre fitxers índex en el cor de la base de dades nativa:

- collections.dbx administra la col·lecció jeràrquicament.
- dom.dbx recopila nodes en un arxiu paginat i "associates unique node identifiers to the actual nodes".
- elements.dbx indexa element i atributs.
- words.dbx manté un camí a les aparicions de paraules i es usat per les extensions de cerca fulltext.

Tots els índex estan basats en arbres B+-. Un important punt a tenir en compte es que els índex per element atributs i paraules clau són organitzats per col·lecció i no per

document, és a dir, un element en una col·lecció serà guardat com un simple índex d'entrada en el índex d'elements. Això manté el nombre de pàgines de l'arbre B+- petit i proporciona un millor funcionament per les consultes en una col·lecció sencera.

El fitxer índex collections.dbx administra la col·lecció jeràrquicament i mapeja noms de col·lecció a objectes de col·lecció. Degut a consideracions de funcionament, les descripcions de documents són sempre guardades amb el objecte col·lecció al qual pertany. Un únic identificador és assignat a cada col·lecció i document durant la indexació.

El magatzem de dades (dom.dbx) representa el component central de l'arquitectura d'emmagatzematge de eXist. Consisteix en un fitxer paginat en el qual tot node document és guardat d'acord amb el model d'objectes (DOM). El magatzem de dades és recolzat per un la B+-Tree multi-arrel en el mateix arxiu per a associar els identificadors de node únic d'alt nivell en un document donat a la direcció d'emmagatzematge del node a la secció de dades.

Només els elements del nivell superior són indexats per un arbre B+-. Els atributs, nodes text i elements a nivells inferiors del node document són solament escrits a les pàgines de dades sense afegir una clau en l'arbre B+-. L'accés a aquesta classe de nodes és proporcionada creuant el més proper ancestre disponible trobat a l'arbre. De totes maneres els accessos directes a aquests nodes són molt poc freqüents. La màquina de consulta processa la majoria de tipus de expressions XPath sense accedir a dom.dbx

6.2 MODES DE TREBALL AMB EXIST

Si ens interessa crear les nostres pròpies aplicacions que accedeixin al servidor d'aquesta BD i a les diferents col·leccions que pot contenir ho podem fer de diverses maneres:

a) Creant aplicacions web que treballin amb XQuery.

En aquest cas hi ha dos opcions per generar directament pàgines web partint de consultes XQuery.

1. Aplicacions web amb XQueryServlet: En aquest cas eXist, a través d'una adreça web patró que es pot configurar a l'arxiu de configuració "web.xml" i unes pàgines que han de tenir una terminació concreta per que siguin reconegudes, agafa la resta de l'adreça que va darrera del patró i ho executa tractant-lo com una consulta adreçada al servidor de BD, i el resultat obtingut el retorna cap al client que li ha fet la comanda.

Les pàgines que porten les consultes han de tenir una terminació ".xql" per poder ser identificades. Aquesta pàgina pot contenir XML i XQuery de manera que abans de ser retornada al client se substitueixen les parts de codi XQuery pel resultat de la consulta realitzada a la BD.

Amb això tenim una pàgina web dinàmica que s'actualitza amb les consultes que se li poden fer en línia.

2. Aplicacions amb XQueryGenerator: Usant un generador com per exemple [Cocoon](#) que treballa com el servlet, però on el resultat és posteriorment tractat abans de ser retornat al client.

b) Amb l'API web HTTP: REST.

Aquest API proporciona unes opcions de treball que estan directament associades sobre els mètodes de consulta clàssics de les pàgines web. Aquests mètodes són: GET, PUT, DELETE i POST. El seu funcionament és molt semblant a com funcionen aquestes opcions als servidors web.

Aquesta opció de treball és la més senzilla i ràpida d'accedir directament al servidor de la BD eXist, però també és la més restringida de totes. Les consultes que podem fer utilitzant aquest mètode utilitzen com a URL una adreça web que conté un patró d'adreça per localitzar les col·leccions amb què es vol treballar, i a continuació en aquesta mateixa adreça, com opcions, se li afegeixen tant la consulta que es fa com altres opcions, tal com es mostren a la taula 5.

Opció	Explicació
_xsl=fulla_d'estil_XSL	La fulla d'estil que se li aplicarà al resultat al retornar
_query=expressió XPath/XQuery	L'expressió amb la consulta que es vol fer

_indent=yes no	Retorna el codi amb tabulacions segons els elements
_encoding=la codificació	La codificació que se li aplicarà al resultat
_howmany=número d'elements	El número d'elements que volem que retorni
_start=punt d'inici	L'element d'inici de retorn
_wrap=yes no	Retorna el resultat embolicat per un element concret

Taula 2.Opcions de consulta

c) Escrivint aplicacions java utilitzant l'API XML:DB.

Aquest seria el mètode més complet, encara que també el més complicat degut a que tractaríem amb el API java. Aquesta API conté les funcions per accedir a tots els aspectes de la BD. Per tant podríem entrar directament a la manipulació de les col·leccions i els documents XML. Això significaria fer programes java que fessin les mateixes funcions que la BD.

d) Utilitzant l'API XML-RPC

Aquest API proveeix un mètode simple per cridar procediments remots des de gran quantitat de llenguatges, per exemple PHP, JSP, etc.

e) Utilitzant Web Services: SOAP

SOAP representa una alternativa a la utilització del API XML-RPC, ja que mentre que amb XML-RPC les crides a les funcions s'han de fer completament a mà, amb SOAP moltes de les crides a les funcions de baix nivell ja es generen automàticament.

Com a inconvenient de SOAP s'ha de dir que les eines SOAP són més difícils d'utilitzar, ja que estan dins l'àmbit dels Web Services. En aquest cas, eXist ofereix dos serveis web, un per fer consultes al servidor i demanar documents, i un altre per afegir i eliminar documents i col·leccions.

CAPÍTOL 7. EL LENGUATGE DE CONSULTA XQuery**7.1 DEFINICIÓ DE XQuery.**

Podem dir que XQuery és a XML el mateix que SQL és a la base de dades relacionals. XQuery és un llenguatge de consulta dissenyat per a escriure consultes sobre col·leccions de dades expressades en XML. Abasta des d'arxius XML fins a bases de dades relacionals amb funcions de conversió de registres a XML. La seva principal funció és extreure informació d'un conjunt de dades organitzades com un arbre nari d'etiquetes XML. En aquest sentit XQuery és independent de l'origen de les dades.

XQuery és un llenguatge funcional, el que significa que, en comptes d'executar una llista de comandaments com un llenguatge procedimental clàssic, cada consulta és una expressió que és avaluada i retorna un resultat, igual que en SQL. Diverses expressions poden combinar-se d'una manera molt flexible amb altres expressions per a crear noves expressions més complexes i de major potència semàntica.

XQuery està cridat a ser el futur estàndard de consultes sobre documents XML. Actualment, XQuery és un conjunt d'esborranys en el qual treballa el grup W3C. No obstant això, malgrat no tenir una redacció definitiva ja existeixen o estan en procés nombroses implementacions de motors i eines que ho suporten.

7.2 REQUERIMENTS TÈCNICS DE XQuery.

El grup de treball en XQuery del W3C ha definit un conjunt de requeriments tècnics per a aquest llenguatge. Els més importants es detallen a continuació.

- XQuery ha de ser un llenguatge declaratiu. Igual que SQL cal indicar el que es vol, no la manera d'obtenir-lo.
- XQuery ha de ser independent del protocol d'accés a la col·lecció de dades. Una consulta en XQuery ha de funcionar igual al consultar un arxIU local que al consultar un servidor de bases de dades que al consultar un arxIU XML en un servidor web.
- Les consultes i els resultats han de respectar el model de dades XML.
- Les consultes i els resultats han d'oferir suport per als namespace .
- Ha de ser capaç de suportar XML-Schemas i DTD's i també ha de ser capaç de treballar sense cap d'ells.
- XQuery ha de poder treballar amb independència de l'estructura del document, això és, sense necessitat de conèixer-la.
- XQuery ha de suportar tipus simples, com sencers i cadenes, i tipus complexos, com un node compost per diversos nodes fills.

- Les consultes han de suportar quantificadors universals (per a tot) i existencials.
- Les consultes han de suportar operacions sobre jerarquies de nodes i seqüències de nodes.
- Ha de ser possible en una consulta combinar informació de múltiples fonts.
- Les consultes han de ser capaces de manipular les dades independentment de l'origen d'aquests.
- Mitjançant XQuery ha de ser possible definir consultes que transformin les estructures d'informació originals i ha de ser possible crear noves estructures de dades.

El llenguatge de consulta ha de ser independent de la sintaxi, això vol dir que, ha de ser possible que existeixin diverses sintaxi diferents per a expressar una mateixa consulta en XQuery. Encara que XQuery i SQL es puguin considerar similars en gairebé la totalitat dels seus aspectes, el model de dades sobre el qual es sustenta XQuery és molt diferent del model de dades relacional sobre el qual es sustenta SQL, ja que XML inclou conceptes com jerarquia i ordre de les dades que no estan presents en el model relacional.

Per exemple, a diferència de SQL, en XQuery l'ordre és que es trobin les dades, és important i determinant, ja que no és el mateix buscar una etiqueta dintre d'una etiqueta que totes les etiquetes del document (que poden estar niades dintre d'una etiqueta o fora). XQuery ha estat construït sobre la base de Xpath.

Xpath és un llenguatge declaratiu per a la localització de nodes i fragments d'informació en arbres XML. XQuery es basa en aquest llenguatge per a realitzar la selecció d'informació i la iteració a mitjançant el conjunt de dades.

7.3 CONSULTES EN XQuery.

Una consulta en XQuery és una expressió que llegeix una seqüència de dades en XML i retorna com resultat una altra seqüència de dades en XML. Un detall important és que, a diferència del que succeeix en SQL, en XQuery les expressions i els valors que retornen són dependents del context.

Per exemple els nodes que apareixeran en el resultat depenen dels namespaces, de la posició on aparegui l'etiqueta arrel del node (dintre d'una altra, per exemple), etc. En XQuery les consultes poden estar compostes per clàusules de fins a cinc tipus diferents.

Les consultes segueixen la norma FLWOR (llegit com flower), sent FLWOR les sigles de For, Let, Where, Order i Return.

A continuació, en la taula 6, es descriu la funció de cada bloc:

For	Vincula una o més variables a expressions escrites en XPath, creant un flux de tuples en el qual cada tupla està vinculada a una de les variables.
Let	Vincula una variable al resultat complet d'una expressió afegint aquests vincles a les tuples generades per una clàusula for o, si no existeix cap clàusula for, creant una única tupla que contingui aquests vincles.
Where	Filtra les tuples eliminant tots els valors que no compleixin les condicions donades.
Order by	Ordena les tuples segons el criteri donat.
Return	Construeix el resultat de la consulta per a una tupla donada, després d'haver estat filtrada per la clàusula where i ordenada per la clàusula order by.

Taula 3. Possibles clàusules en una consulta XQuery

En XQuery, el terme tupla, ens estem referint a cadascun dels valors que pren una variable. A continuació, en la figura 4, es mostra gràficament l'ordre que s'executa cada clàusula d'una consulta i els resultats de cadascuna:

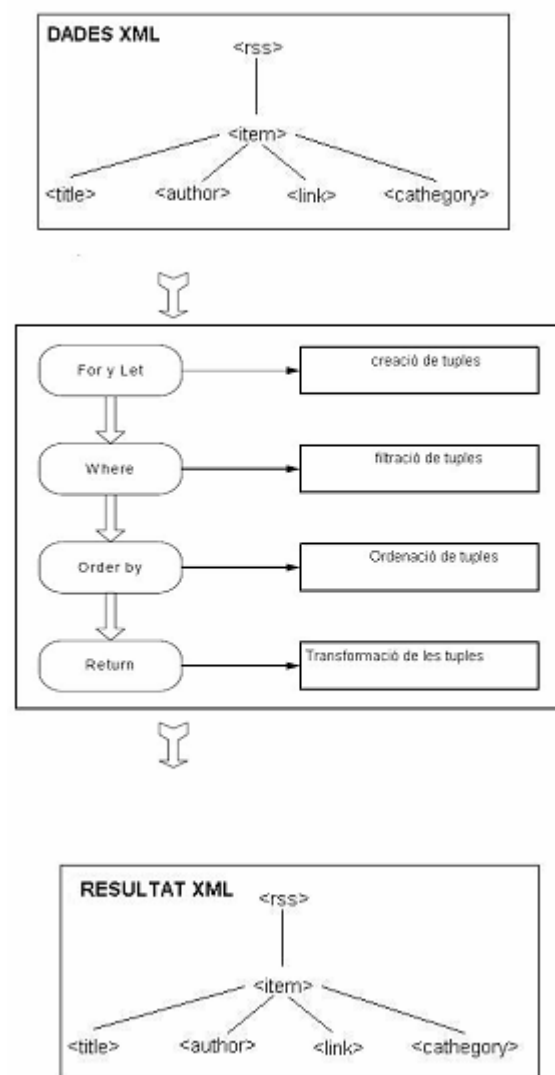


Figura 3.Ordre d'execució i resultat de les cinc clàusules possibles

7.3.1 REGLES GENERALS.

A continuació, enunciamem una sèrie de regles que ha de complir qualsevol consulta escrita en XQuery:

- For i let serveixen per a crear les tuples amb les quals treballarà la resta de les clàusules de la consulta i poden usar-se tantes vegades com es desitgi en una consulta, fins i tot dintre d'altres clàusules. No obstant això solament pot declarar-se una única clàusula where, una única clàusula order by i una única clàusula return.
- Cap de les clàusules FLWOR és obligatòria en una consulta XQuery. Per exemple, una expressió XPath, com la qual es mostra a continuació, és una consulta vàlida i no conté cap de les clàusules FLWOR.

- És possible especificar diversos criteris d'ordenació en la clàusula `order by`, separant-los per comes. Els criteris d'ordenació s'apliquen per ordre d'esquerra a dreta.

7.4 FUNCIONS D'ENTRADA

XQuery utilitza les funcions d'entrada en les clàusules `for` o `let` o en expressions XPath per a identificar l'origen de les dades. Actualment l'esborrany de XPath i XQuery defineix dues funcions d'entrada diferents, `doc(URI)` i `collection(URI)`.

La funció `doc(URI)` retorna el node document, o node arrel, del document referenciat per un identificador universal de recursos (URI). Aquesta és la funció més habitual per a accedir a la informació emmagatzemada en arxius.

La funció `collection(URI)` retorna una seqüència de nodes referenciats per una URI, sense necessitat que existeixi un node document o node arrel. Aquesta és la funció més habitual per a accedir a la informació emmagatzemada en una base de dades que tingui capacitat per a crear estructures de dades XML.

7.5 EXPRESSIONS CONDICIONALS.

A més de la clàusula `where`, XQuery també suporta expressions condicionals del tipus "if-then-else" amb la mateixa semàntica que en els llenguatges de programació més habituals (C, Java, Delphi, etc..).

La clàusula `where` d'una consulta permet filtrar les tuples que apareixeran en el resultat, mentre que una expressió condicional ens permet crear una o altra estructura de nodes en el resultat que depengui dels valors de les tuples filtrades. A diferència de la majoria dels llenguatges, la clàusula `else` és obligatòria i ha d'aparèixer sempre en l'expressió condicional. El motiu d'això és que tota expressió en XQuery ha de retornar un valor. Si no existeix cap valor a retornar al no complir-se la clàusula `if`, retornem una seqüència buida amb `'else ()'`,

7.6 QUANTIFICADORS EXISTENCIALS

XQuery suporta dos quantificadors existencials cridats "some" i "every", de tal manera que ens permet definir consultes que retorni algun element que satisfaci la condició

("some") o consultes que retornin els elements en els quals tots els seus nodes satisfacin la condició ("every").

7.7 OPERADORS I FUNCIONS PRINCIPALS.

El conjunt de funcions i operadors suportat per XQuery 1.0 és el mateix conjunt de funcions i operadors utilitzat en XPath 2.0 i XSLT 2.0. XQuery suporta operadors i funcions matemàtiques i de cadenes, per al tractament d'expressions regulars, comparatives de dates i hores, manipulació de nodes XML, manipulació de seqüències, comprovació i conversió de tipus i lògica booleana. A més permet definir funcions pròpies i funcions dependents de l'entorn d'execució del motor XQuery. Els operadors i funcions més importants es mostren en la taula 7.

Matemàtics	+, -, *, div(*), idiv(*), mod
Comparació	=, !=, <, >, <=, >=, not().
Seqüència	union (), intersect, except
Arrodoniment	floor(), ceiling(), round()
Funcions d'agrupació	count(), min(), max(), avg(), sum()
Funcions de cadenes	concat(), string-length(), startswith(), ends-with(), substring(), upper-case(), lower-case(), string()
Ús general	distinct-values(), empty(), exists()

Taula 4. Operadors i funcions en XQuery

La divisió s'indica amb l'operador 'div' ja que el símbol '/' és necessari per a indicar camins.

L'operador 'idiv' és per a divisions amb enters en els quals s'ignora la resta. El resultat d'un operador aritmètic en el qual un, o dos dels operants sigui una cadena buida és una cadena buida. Com regla general el funcionament de les cadenes buides en XQuery és anàleg al funcionament dels valors nuls en SQL.

L'operador unió rep dues seqüències de nodes i retorna una seqüència amb tots els nodes existents en les dues seqüències originals.

L'operador d'intersecció rep dues seqüències de nodes com operands i retorna una seqüència contenint tots els nodes que apareguin els dos operands.

L'operador de substracció (except) rep dues seqüències de nodes com operands i retorna una seqüència contenint tots els nodes del primer operant que no apareguin en el segon operant.

La funció distinct-values() extreu els valors d'una seqüència de nodes i crea una nova seqüència amb valors únics, eliminant els nodes duplicats.

La funció empty() retorna cert quan l'expressió entre parèntesi està buida.

7.8 COMENTARIS.

Els comentaris en XQuery, a diferència de XML, van tancats entre cares somrients, tal com es mostra a continuació.

(: Això és un comentari:)

CAPÍTOL 8. EXEMPLES DE CONSULTES XQuery A eXist

Un cop engegat el servidor eXist, amb els scripts que ja porta definits, podem executar el client de la BD, implementat també en java.. En un primer moment, al accedir al client ens demana un usuari (admin) i una contrasenya (cadena buida) -Figura 5- per accedir, i a més quin és el directori base des del que penjaran les diferents col·leccions.

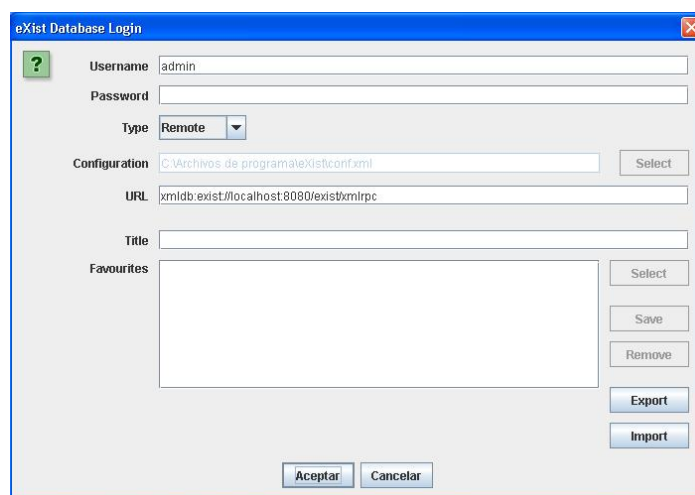


Figura 4. eXist Database Login

Des del client, de la mateixa manera que des de la interface web (en mode servlet), podem treballar sobre col·leccions -figura 6-, i disposa també d'una pantalla des de la que es poden fer consultes a les col·leccions instal·lades. En aquest projecte es treballarà en el directori TFC.

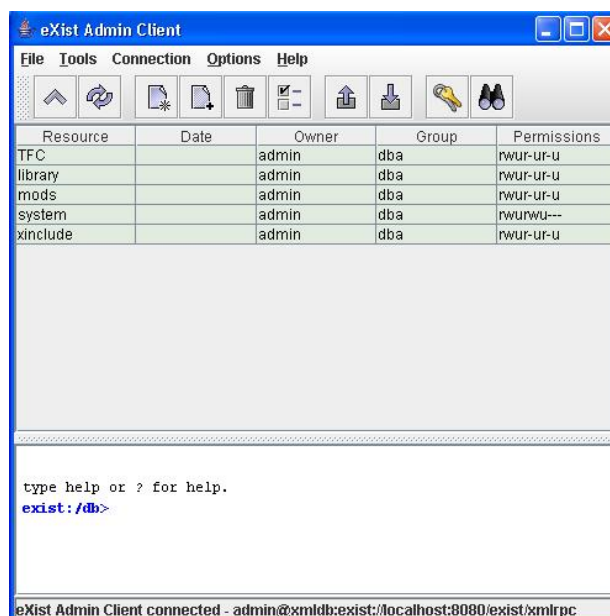


Figura 5. eXist Admin Client

PRIMER EXEMPLE DE CONSULTA

Aquesta consulta resulta útil per llistar tots els títols de llibres que tenim en el document rss i que ens els presenti ordenat alfabèticament.

```
for $b in doc("rss0.xml")//item
order by $b/title
return $b/title
```

Les barres: "/" no indiquen comentaris, sinó que són part de l'expressió que indica la localització dels valors que prendrà la variable \$b. La expressió FLWOR vincula variables a valors amb clàusules for i utilitza aquest vincle per a crear noves estructures de dades XML. La consulta ordena alfabèticament (order by) per <title> i retorna (return) tots els <title> amb el seu contingut.

SEGON EXEMPLE DE CONSULTA

Aquesta consulta resulta útil per llistar tots els autors de llibres que tenim en el document rss i que ens els presenti ordenat alfabèticament.

```
for $b in doc("rss0.xml")//item
```

```
order by $b/author
return $b/author
```

El funcionament de la consulta es el mateix que per title però aplicat a la etiqueta <author>.

TERCER EXEMPLE DE CONSULTA

Aquesta consulta retorna cada títol de llibre amb el seu autor que tenim en la base de dades.

```
for $t in doc("rss0.xml")//item
return
<item>
  {$t/title}
  {$t/author}
</item>
```

QUART EXEMPLE DE CONSULTA

Aquesta consulta ens permet la recerca d'un títol en la base de dades mitjançant una paraula clau, en el exemple, s'ha fet la recerca dels títols que continguin la paraula "Màquines" i solament s'ha trobat un.

```
for $b in doc("rss0.xml")//item
where some $p in $b//title satisfies
(contains($p,"Màquines"))
return $b/title
```

CINQUÈ EXEMPLE DE CONSULTA

Aquesta consulta retorna els títols dels llibres d'un autor en concret, en el exemple hem fet la recerca dels llibres del autor "Sanjurjo" i ens ha retornat un llibre.

```
for $b in doc("rss0.xml")//item
where some $p in $b//author satisfies
(contains($p,"Sanjurjo"))
return $b/title
```

SISÈ EXEMPLE DE CONSULTA

Aquesta consulta és molt similar al exemple anterior, el que fa és trobar una paraula clau en els títols que tenim a la base de dades, i ens torna els autors que tenen llibres amb els quals el seu títol conté la paraula clau.

```
for $b in doc("rss0.xml")//item
where some $p in $b//title satisfies
(contains($p,"Principis"))
return $b/author
```

CAPÍTOL 9. CAS PRÀCTIC: APLICACIÓ WEB PER CONSULTAR DOCUMENTS RSS

Com a cas d'ús, de tot el que s'ha tractat fins ara, toca fer una aplicació senzilla, que permeti fer cerques sobre documents RSS emmagatzemats a una base de dades XML nativa. La BD escollida ha estat eXist, ja que ha mostrat un funcionament molt bo en totes les proves que s'han realitzat.

L'aplicació dona una idea bastant clara de com totes les tecnologies esmentades fins ara poden treballar juntes.

El sistema consta de dos parts:

Un ordinador amb Windows NT, aquest ordinador és el que realment ha suportat tot el treball de programació i s'han instal·lat part de les eines necessàries per fer aquest treball, que han estat:

- El intèrpret de PHP.
- El servidor web Apache.
- Bloc de notes amb les que s'han fet les pàgines web necessàries.
- La part de consultes a la BD:eXist s'ha fet mitjançant amb l'API web:REST

Un altre ordinador, amb sistema operatiu Windows NT, en el que s'ha instal·lat:

- La BD eXist en la seva versió de treball com a servlet concretament la versió "eXist-1.0".
- Els documents RSS s'han tret de la direcció: <http://www.amazon.com/exec/obidos/subst/xs/syndicate.html/104-7112976-5718346>. els quals han estat modificats manualment per aconseguir uns exemples de consultes més didàctics.

9.1 COMPONENTS DE L'APLICACIÓ

Aquesta aplicació consta de dues parts que estan definides per les diferents pantalles pel treball interactiu amb la base de dades i els documents que conté. Les diferents parts de que es compona són:

- Una pantalla d'inici en la que es mostra tant el títol de l'aplicació, el nom del alumne que ha realitzat el TFC, el nom del consultor encarregat d'aquest TFC i els estudis als quals correspon, la qual es mostra en la Figura 14.

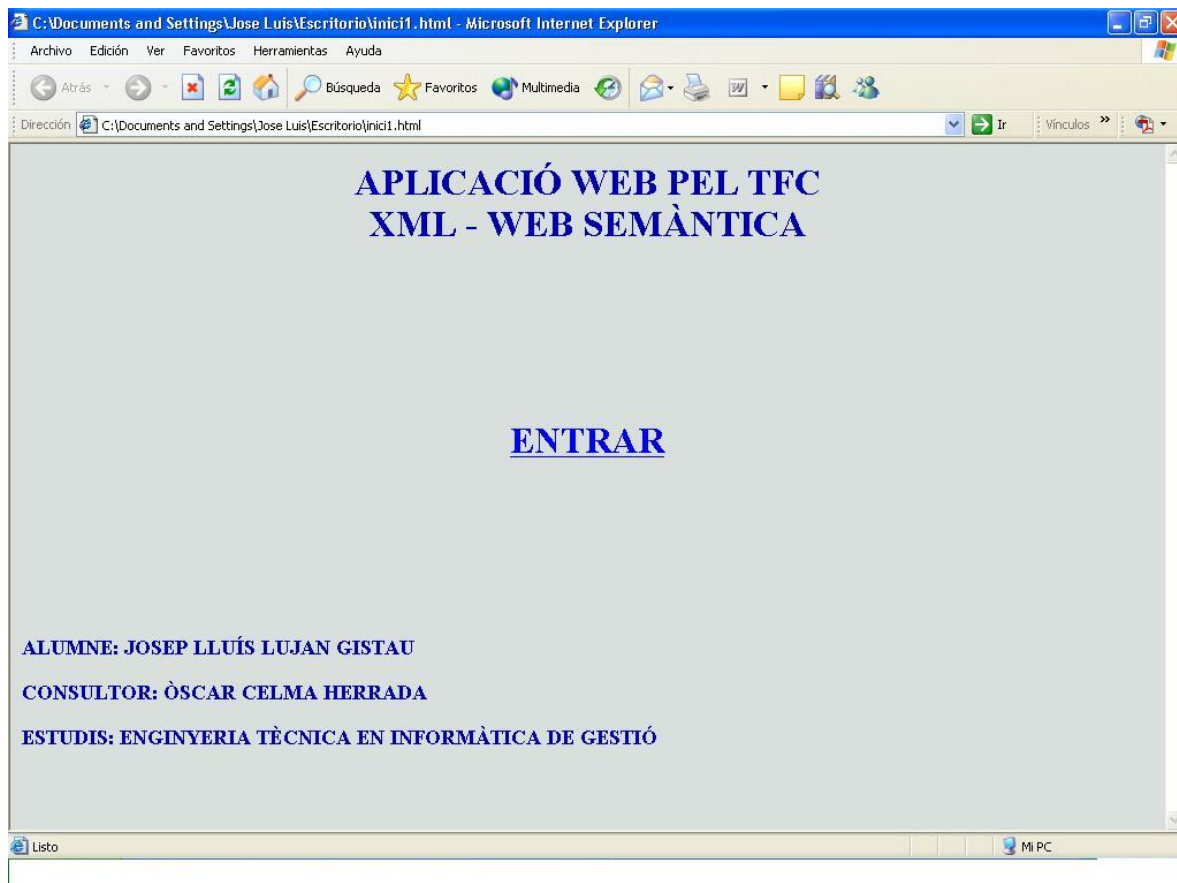


Figura 6. Pantalla de presentació de la aplicació Web

- Una segona pantalla que esta dividida en dos **FRAMES**, els quals mostren a la esquerra les diferents opcions de recerca i a la dreta el frame on mostrarem els resultats de la consulta mitjançant un **IFRAME** tal com es mostra a la figura 15.

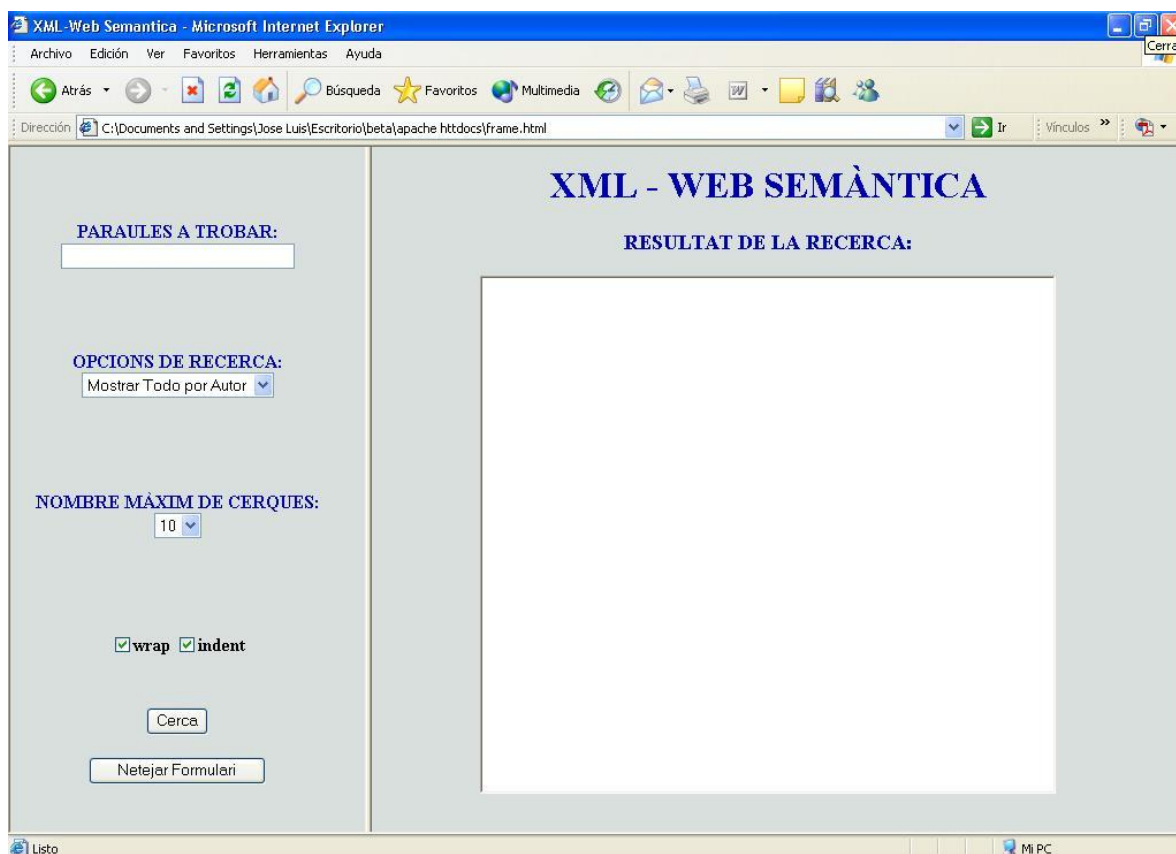


Figura 7. Formulari de Recerca i mostra de resultats

Seguidament podem veure en detall les diferent opcions de recerca que té el formulari:

- Opcions de recerca on s'han posat els sis tipus de consultes que es poden fer en l'arxiu RSS tal com es mostren a la Figura 16, podent triar qualsevol opció a l'hora de fer la cerca.



Figura 8. Opcions de recerca

- Numero màxim de consultes que volem que es representi en el Iframe de sortida tal com es mostra a la Figura 17.



Figura 9. Número màxim de recerques

- Les paraules per la tria de coincidències es posaran dintre de la caixa de text de la figura 18.

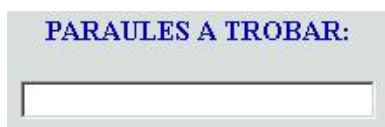


Figura 10. Caixa de text de recerca

- Els atributs wrap e indent de tipus booleà fan referència al tipus de formateig del text de sortida per exemple que el text ingressat no pot superar els marges laterals, és a dir, quan l'usuari arribi al final del marge dret automàticament desplaçarà el cursor cap avall sense esperar que l'usuari presioni la tecla ENTER, aquestes opcions es mostren en la figura 19.

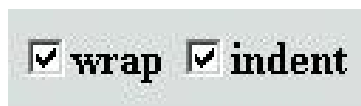


Figura 11. Opcions wrap i indent

- En la figura 20 es mostra els botons de "cerca" i "Netejar Formulari" ens permeten llançar la consulta i buidar la caixa de text de paraules a trobar per realitzar un altre cerca.



Figura 12. Botons Cerca i Neteja de Formulari

- En la figura 21 es mostra el IFRAME de sortida on es representen els resultats de la recerca en forma de taula.

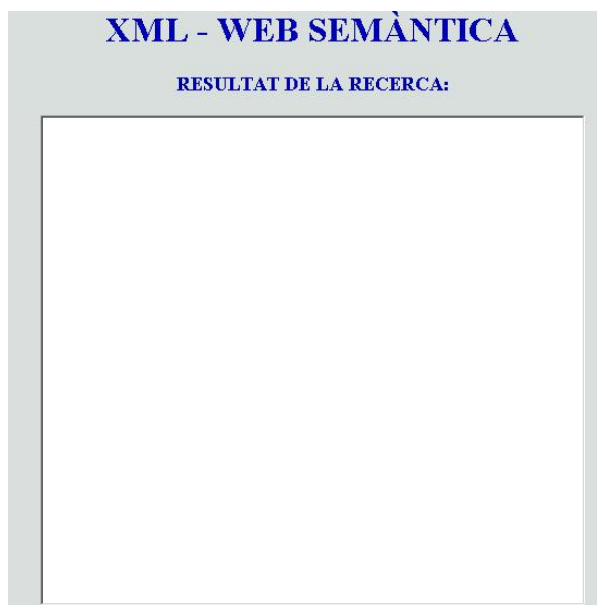


Figura 13. IFrame de mostra de resultats

9.2 EXEMPLES DE CONSULTA

Seguidament es presenta uns exemple de consultes en la pagina Web amb els seus resultats.

- “Mostrar tot per Autor:” el resultat es un llistat de parelles autor títol de cada llibre ordenats en ordre alfabètic per l'autor del llibre tal i com es pot veure a la figura 22.

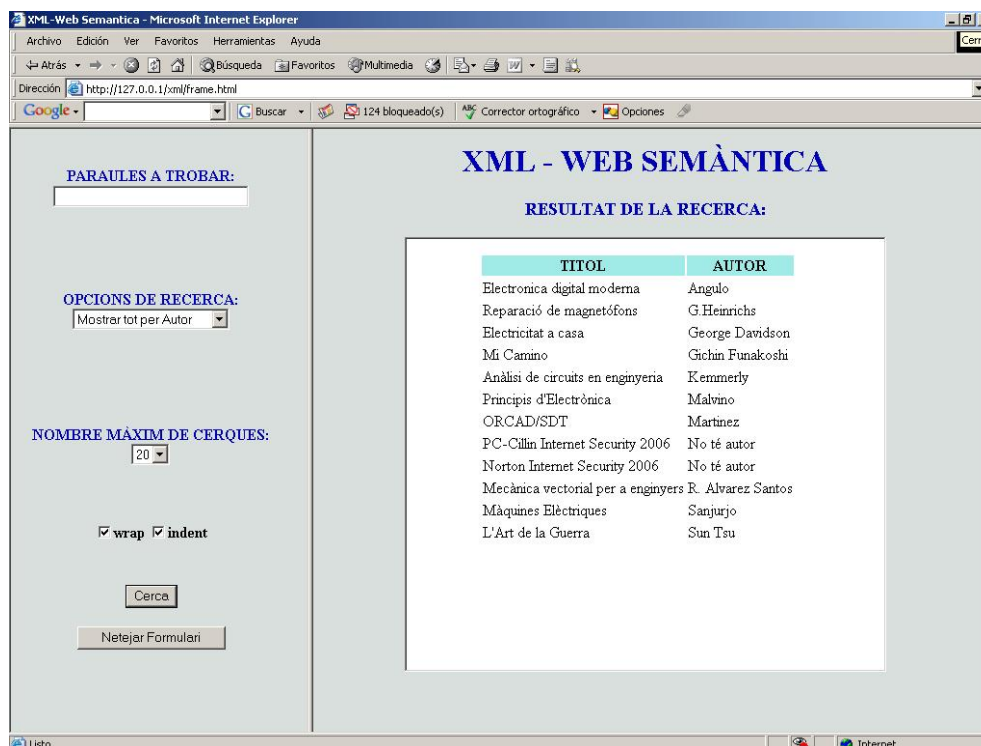


Figura 14. Consulta Mostrar tot per autor

- “Mostrar tot per Títol”: molt semblant a l’anterior a diferència es que ens mostra el resultat de les parelles autor – títol ordenades alfabèticament pel títol del llibre tal com es mostra en la figura 23.

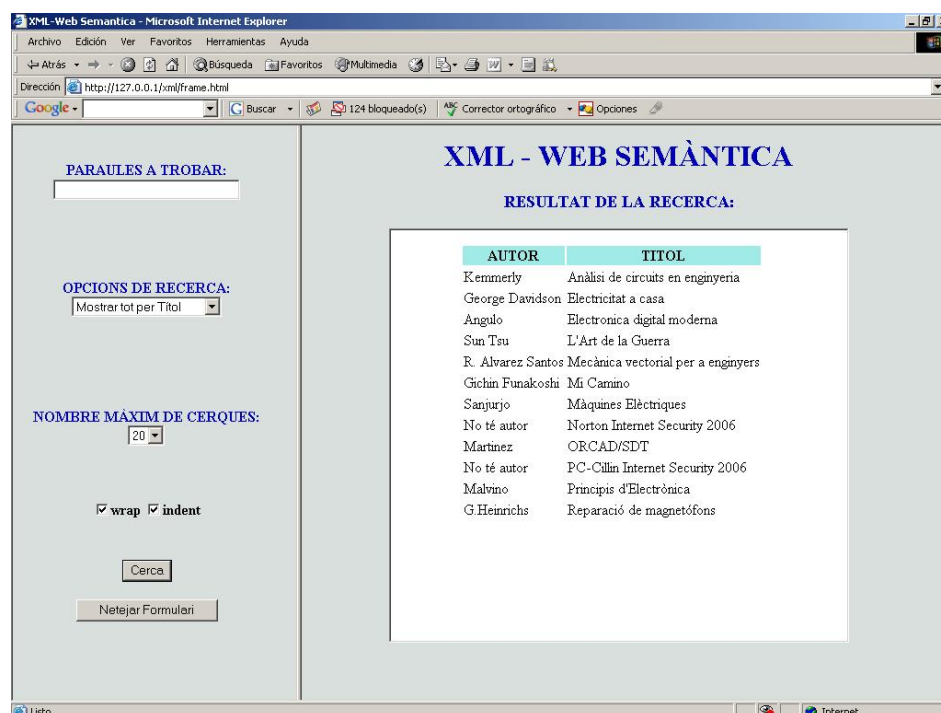


Figura 15. Consulta Mostrar tot per títol

- “Mostrar tots els autors”: Ens mostra tots els autors de la base de dades per ordre alfabètic, en la figura 24 podem veure un exemple d’aquests tipus de recerca.

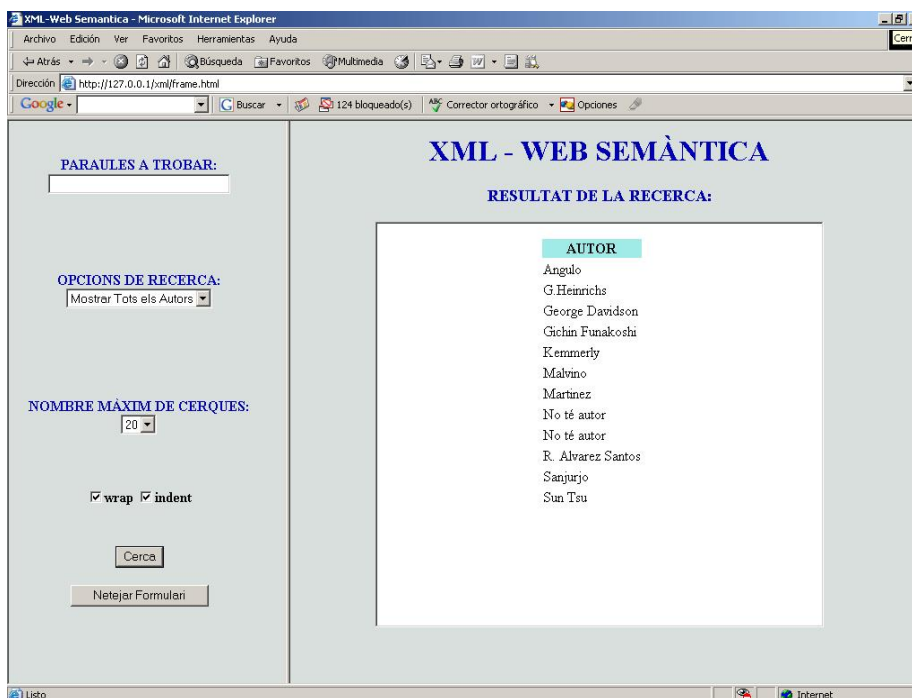


Figura 16. Consulta Mostrar tots els autors

- “Mostrar tots els títols”: Ens mostra tots els títols de la base de dades per ordre alfabètic, en la figura 25 podem veure un exemple d’aquets tipus de recerca.

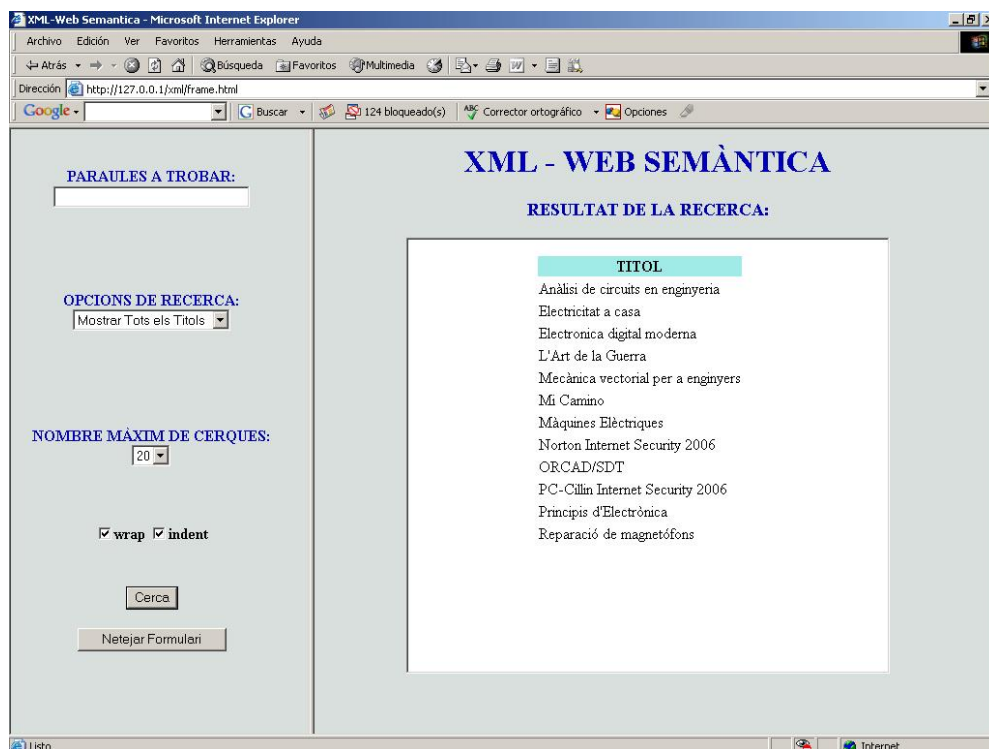


Figura 17. Consulta Mostrar tots els títols

- “Cerca per autors”: El usuari pot cercar un autor per consultar si està a la base de dades i ens mostra, si troba les parelles títol – autor. En el exemple de la figura s’ha fet la cerca de l’autor Sun Tsu , trobant la parella L’art de la guerra – Sun tsu tal com es mostra a la figura 26.

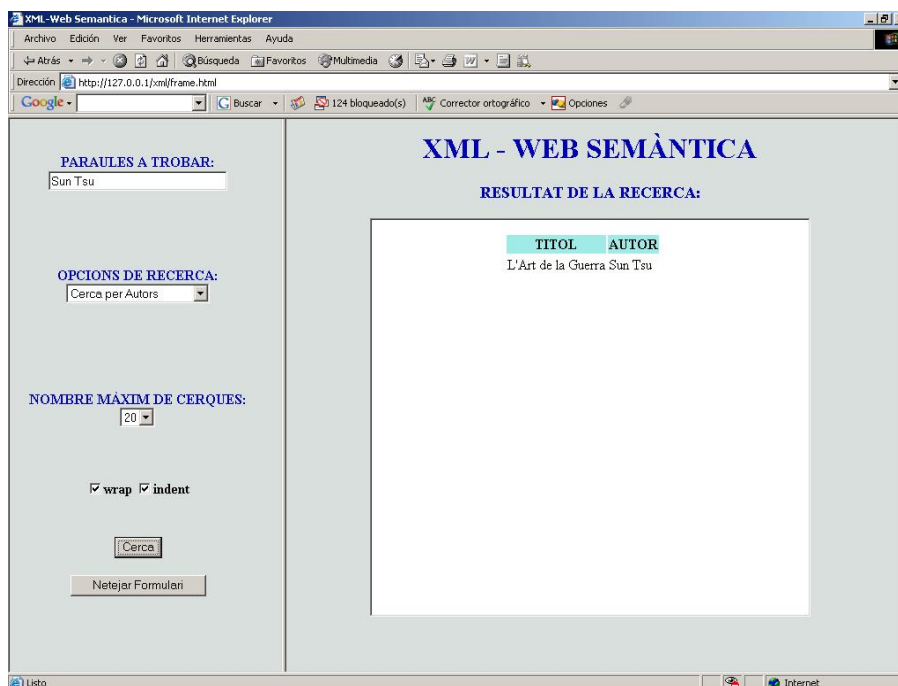


Figura 18. Consulta Cerca per autors

- “Cerca per títols”: El usuari pot cercar un títol o una paraula que porti el títol per consultar, si aquest títol o paraula es troba a la base de dades ens mostrarà la parella autor - títol. En el exemple de la figura s’ha fet la cerca per la combinació “Electr” per trobar en el títol, i s’han trobat les tres parelles autor – títol que es mostren a la figura 27.

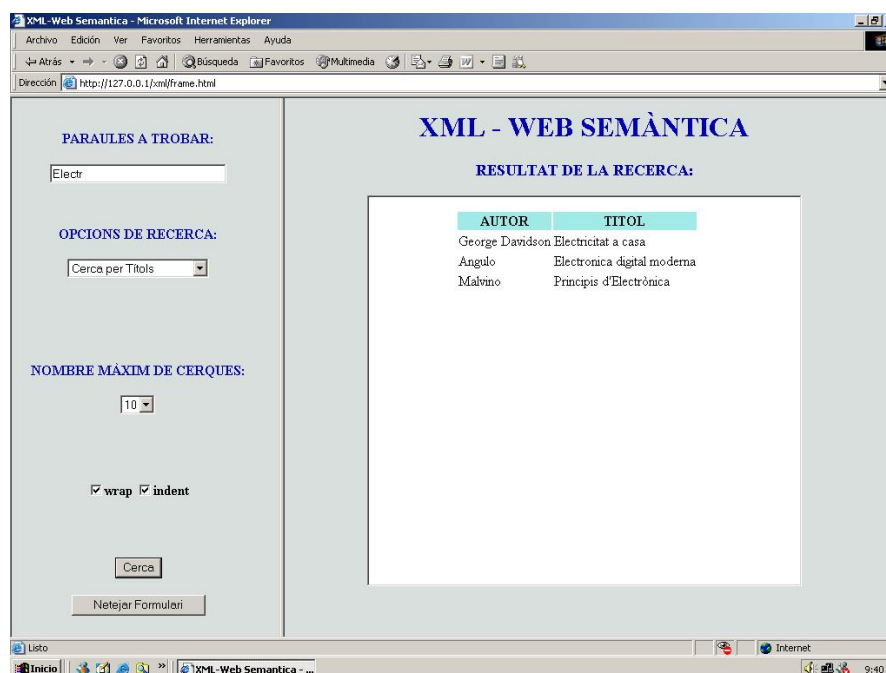


Figura 19. Consulta Cercar per títols

CAPÍTOL 10. CONCLUSIONS

La principal conclusió que caldria esmentar és que s'han aconseguit els objectius proposats:

- He obtingut una aplicació que funciona.
- He ampliat notablement els coneixements adquirits durant la carrera, però cal destacar que han estat ampliat amb continguts força actuals con ara fer servir una base dades XML i programar una interface gràfica d'usuari.

Tenint en compte que les assignatures fetes fins ara només abastaven parts del que podríem anomenar un projecte íntegre (l'anàlisi i el disseny d'aplicacions, el disseny d'interfaces gràfiques d'usuari, la programació o implementació de codi, el disseny i la gestió de base de dades...) cal destacar la importància d'aquest TFC com a element aglutinador de coneixements perquè m'ha permès desenvolupar una aplicació seguint totes les fases d'un projecte.

Ha quedat patent que en el desenvolupament de qualsevol projecte seriós, ha de dotar-se a l'etapa de recollida d'informació i anàlisi d'un temps prou ampli per garantir l'èxit de les següents fases del projecte

Una altra conclusió per a aquest treball és que personalment l'he trobat molt interessant degut al fet que he pogut fer una investigació i recopilació d'informació que fa tan sols uns anys haurien estat molt més complicades, ja que mitjançant l'ús de les noves tecnologies com són Internet i tot allò que l'envolta, aquestes cerques d'informació es tornen molt més senzilles que si les hagués hagut de fer anant a biblioteques o cercant llibres que continguessin la informació que he necessitat.

En quant als llenguatges de consultes i sistemes nadius, sembla que el futur dels llenguatges de consulta sobre XML i els sistemes nadius no amaga, però, certa imatge de provisionalitat, derivada tant de les constants propostes d'innovació, en el que respecta a les especificacions, fetes pels diferents grups de treball del W3C en el que respecta a l'XQuery

Un altre punt que m'ha semblat molt interessant i que no coneixia eren els estàndards RSS / Atom i el seu intent mantenir actualitzades les pàgines web mitjançant la sindicació de continguts web. Aquestes noves tecnologies crec que són un pas lògic

degut a les revolucions constants per part de les empreses en el entramat d'Internet de mantenir actualitzada la informació que se'ls mostra als usuaris.

CAPITOL 11. GLOSSARI

Atom: Format XML per la sindicació de continguts Web.

Document ben format: Document de text ASCII que segueix les recomanacions d'XML.

Document vàlid: Document XML que segueix les especificacions d'una DTD.

DTD (Document Type Definition, Definició de tipus de document): Declaració en un document XML que especifica les restriccions en la seva estructura.

HTML (Hyper Text Markup Language, llenguatge de marcatge d'hipertext): llenguatge de programació Web format per un conjunt d'etiquetes que serveixen per definir la forma de presentació del text i altres elements d'una pàgina Web.

Metadades: Dades que tracten sobre altres dades. Informació sobre la informació. Les metadades descriuen les característiques, continguts informatius, gestió de drets i accessibilitat als recursos d'informació.

Metallenguatge: llenguatge que permet descriure altres llenguatges.

Namespace (espai de noms): Un namespace XML és una col·lecció de noms, identificats per una referència URI, que són fets servir en documents XML com noms d'elements o d'atributs. Això permet que els noms no siguin utilitzats de forma ambigua, facilitant el fet de compartir semàntiques

Parsejar: Recórrer un fitxer per extreure la seva informació.

Sindicació: Publicació.

RSS (Real Simple Syndication): Format XML per la sindicació de continguts Web.

URL (Unified Resource Locator): Localitzador de recursos unificat. En Internet és una comanda que indica al navegador on es troba un recurs determinat i quin tipus d'informació conté.

XML (eXtended Markup Language): llenguatge de marques, l'objectiu del qual és representar l'estructura de la informació amb independència de la mateixa informació. És independent de la plataforma sobre la qual treballa.

CAPÍTOL 12. BIBLIOGRAFIA

- Servidor Web de Apache versió 1.3.35: <http://www.apache.org>
- PHP: <http://www.php.net>

- Amazon: www.amazon.com
<http://www.amazon.com/exec/obidos/subst/xs/syndicate.html/104-7112976-5718346>
- World Wide Web Consortium, "W3C": <http://www.w3.org/>
- XQuery 1.0 and XPath 2.0 Functions and Operators: <http://www.w3.org/tr/XQuery-operators/>
- XUpdate i el seu DTD associat: <http://xmldb.org.sourceforge.net/xupdate/xupdate-wd.html>
- XML:DB XUpdate Use Cases: <http://www.xmldatabases.org/projects/XUpdate-UseCases/>
- BD eXist: <http://exist.sourceforge.net/>
- PHP en castellà: <http://es2.php.net/manual/es/>
- XML-ES: tutorials sobre XML: <http://www.it.uc3m.es/~xml/indice.html>
- Tota la informació i més...: http://www.google.es/advanced_search?hl=es

ANNEX A

A.1 INSTAL·LACIÓ JAVA

A.1.1 PROCÉS D'INSTAL·LACIÓ.

1. Cal descarregar l'arxiu `j2sdk-1_4_2_06-linux-i586-rpm.bin` des de <http://www.java.com/en/download> , posteriorment cal fer-lo executable per a poder realitzar l'extracció.

```
chmod +x j2sdk-1_4_2_06-linux-i586-rpm.bin
```

2. S'executa l'arxiu `j2sdk-1_4_2_06-linux-i586-rpm.bin`:

```
./j2sdk-1_4_2_06-linux-i586-rpm.bin
```

3. Es llegeix la llicència i es confirma que s'accepten els termes de la mateixa. Posteriorment s'extreu el paquet RPM `j2sdk-1_4_2_06-linux-i586.rpm`.

4. Com a root instal·lar `j2re-1_4_2_04-linux-i586.rpm`:

```
su rpm -Uvh j2sdk-1_4_2_06-linux-i586.rpm
```

5. Es crea l'arxiu `/etc/profile.d/java.sh` on s'afegeix la ruta de binaris de Java 2 (en el meu cas `/usr/java/j2sdk1.4.2_06/bin`).

```
export PATH=$PATH:/usr/java/j2sdk1.4.2_06/bin
```

```
JAVA_HOME="/usr/java/j2sdk1.4.2_06/"
```

```
export JAVA_HOME
```

6. Es fa executable `/etc/profile.d/java.sh`:

```
chmod 755 /etc/profile.d/java.sh
```

7. S'instal·la l'extensió (Plug-in) Java per a Mozilla:

```
ln -s /usr/java/j2sdk1.4.2_06/plugin/i386/ns610/libjavaplugin_oji.so
```

```
/usr/lib/mozilla/plugins/
```

8. En algunes versions del paquet RPM s'inclou un fitxer que mostra una entrada per l'escriptori, però hi ha una errada d'omissió d'aquest fitxer. Aquest és el responsable de que apareguin les preferències de Java en el menú de preferències de l'escriptori. Si aquest existeix cal editar el fitxer

/usr/share/applications/sun_java.desktop i afegir un ";" al final de la línia Categories=Application;Settings, de manera que el contingut seria el següent:

[Desktop Entry]

Name=Java

Comment=Java Control Panel

Exec=/usr/java/j2sdk1.4.2_06/bin/ControlPanel

Icon=/usr/java/j2sdk1.4.2_06/plugin/desktop/sun_java.png

Terminal=0

Type=Application

Categories=Application;Settings;

Si el fitxer no existeix caldrà generar-lo.

9. Tancar totes les aplicacions i després caldrà reiniciar la sessió.

A.1.2 COMPROVACIONS.

Per a comprovar si Java ha estat instal·lat correctament, s'executa el següent des del terminal *which java*

L' anterior deurà retornar que la comanda *java* està a
/usr/java/j2sdk1.4.2_06/bin/java.

S'obre Mozilla i en teclejar *about:plugins*. Premem enter. Apareixerà informació sobre les extensions instal·lades per a Mozilla, i d'entre elles es podrà veure la informació corresponent a l'extensió (Plug-in) Java.

ANNEX B

B.1 INSTAL·LACIÓ D'EXIST

eXist es pot descarregar des de <http://exist-db.org/index.html>. En el present treball s'ha utilitzat la versió 1.0beta2. Es descarrega l'arxiu *eXist-1.0b2-build-1107.jar*

Per a instal·lar cal fer el següent amb l'ordre Java
java -jar eXist-1.0b2.jar

Si eXist a estat instal·lat al directori `/usr/local/exist` per engegar la base de dades mitjançant un terminal de manera manual cal fer:

```
exist/bin/statup.sh
```

Llavors es pot accedir al sistema de gestió de la BD:XML mitjançant la URL

<http://localhost:8080/exist/index.html>

Per a un inici ràpid es pot fer servir el servidor a través d'un *servidor web* que inclou la distribució font (Jetty) , d'aquesta manera no es requereix cap tipus de software addicional. Pel desplegament d'altres opcions cal fer una ullada al document de desplegament del servidor i llegir la guia ràpida d'inici.

B.1.1 COMPROVAR SI EL SERVIDOR ESTÀ ENEGAT

La pàgina principal d'eXist deu ser ara habilitada a :

<http://localhost:8080/exist/index.xml>.

Per veure si l'enginy de base de dades està executant-se es pot fer click a **Server status** o directament a <http://localhost:8080/exist/status>. El servidor donarà informació de la configuració, el directori usat per l'enginy i les bases de dades habilitades. Si està treballant continua més endavant.

B.1.2 ARXIUS INSTAL·LATS

El directori bin conté diferents shell i batch scripts. Si s'instal·la la distribució font o s'usa l'instal·lador, es trobarà el directori bin directament penjant del directori arrel de la distribució .en ell trobem els següents scripts.

startup.sh

Fa el mateix efecte que

```
java -jar start.jar jetty
```

Engega el servidor web Jetty al port 8080. Es localitza mitjançant

<http://localhost:8080/exist/>

shutdown.sh

Fa el mateix que

```
java -jar start.jar shutdown
```

Es fa servir per fer shut-down a del servidor d'eXist. Sempre cal cridar shutdown abans de matar el procés del servidor

server.sh

Fa el mateix que

```
java -jar start.jar server
```

Llença eXist com un procés independent. D'aquesta manera, eXist és solament accessible a través del XMLRPC i una simple interfície HTTP.

client.sh

Fa el mateix que

```
java -jar start.jar client
```

Crida al client Java.

ANNEX C: Document RSS de la BD del cas pràctic

C.1 "rss0.XML"

```
<?xml version="1.0" encoding="UTF-8"?>
  <rss version="0.91">
    <channel>
      <link>
        http://www.amazon.com/exec/obidos/tg/browse/-/229535/1/?rank=%2Bsalesrank%26dev-t=amznRss
      </link>
      <description>Up-to-date listings of Amazon.com's top items
    </description>
    <language>en-us</language>
    <docs>http://www.amazon.com/exec/obidos/subst/xs/syndicate.html</docs>
    <image>
      <title>Amazon.com--Earth's Biggest Selection</title>
      <url>http://images.amazon.com/images/G/01/rcm/logo2.gif
      </url>
      <width>120</width>
```

```
<height>30</height>
<link>http://www.amazon.com</link>
</image>
<item>
  <title>Principis de electronica</title>
  <author>Malvino</author>
  <description>Principis bàsics
d'electronica-- €159.99
  </description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
0B8P8VM/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>Maquines Electriques</title>
  <author>Sanjurjo</author>
  <description>Principis bàsics de maquines
electriques -- €119.99
  </description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
00C0XT1/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>Analisi de circuits en ingenyeria</title>
  <author>Kemmerly</author>
  <description>Important per aprovar la
assignatura</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
0B5I002/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>Electricitat a casa</title>
  <author>Gerorge Davidson</author>
  <description>Un llibre molt bàsic
d'electricitat -- $39.99</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
0AA87JG/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>Reparaciã³ de magnetã³fons</title>
  <author>G.Heinrichs</author>
  <description>totalment obsolet</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
0B8K7SG/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>Norton Internet Security 2006</title>
  <description>Norton Internet Security 2006 --
$59.99</description>
```

```
<link>http://www.amazon.com/exec/obidos/ASIN/B00
0B5NS2G/ref=nosimRss/webservices-20?dev-
t=amznRss</link>
</item>
<item>
  <title>PC-Cillin Internet Security 2006</title>
  <description>PC-Cillin Internet Security 2006 --
  $39.99</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  0BJLN10/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
<item>
  <title>Mecanica vectorial per a
  ingenyers</title>
  <author>R. Alvarez Santos</author>
  <description>Asignatura muy facil</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  01WN16M/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
<item>
  <title>Electronica digital moderna</title>
  <author>Angulo</author>
  <description>El meu llibre preferit de
  electronica digital</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  0CE9IAM/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
<item>
  <title>ORCAD/SDT</title>
  <author>Martinez</author>
  <description>Un dels meus millors
  regals</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  09XB156/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
<item>
  <title>Mi Camino</title>
  <author>Gichin Funakoshi</author>
  <description>Autobiografia</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  09XB156/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
<item>
  <title>El Arte de la Guerra</title>
  <author>Sun Tsu</author>
  <description>El Arte de la Guerra</description>
  <link>http://www.amazon.com/exec/obidos/ASIN/B00
  09XB156/ref=nosimRss/webservices-20?dev-
  t=amznRss</link>
</item>
```

```
</channel>  
</rss>
```

ANNEX D: INSTAL·LACIÓ DE APACHE EN WINDOWS

D.1 DESCÀRREGA DE APACHE 2.0.50

El servidor web Apache es pot descarregar de la pàgina de la Fundació Apache, en Apache.org. En la URL <http://httpd.apache.org/> tenen una secció en particular per al servidor HTTP (Protocol de transmissió utilitzat en la web), en la qual hauríem de buscar l'enllaç per la descàrrega.

Una vegada en la pàgina de descàrrega hem de buscar l'última versió estable o la versió recomanada. El servidor està disponible per a sistemes Unix o Windows, pel que ens oferiran diverses opcions de descàrrega, fins i tot per a obtenir el codi font del programa.

Si desitgem instal·lar-lo en un sistema Windows, necessitarem obtenir l'arxiu Win-32 Binary, que és un instal·lador Windows.

D.1 INSTAL·LACIÓ

Una vegada descarregat, ho hem d'instal·lar en el nostre sistema executant l'arxiu obtingut, després donarà pas a un assistent que ens guiarà en el procés d'instal·lació del servidor, durant el procés ens preguntarà el nom del domini i del servidor, que podem emplenar amb "localhost"

Després ens demanarà també una adreça de correu, que podem emplenar amb qualsevol que desitgem. També ens preguntarà si desitgem que el servidor respongui en el port 80 per a tots els usuaris o si desitgem que només s'activi per a l'usuari actual en el port 8080, quan s'iniciï manualment, el normal és que ho activem per a tots els usuaris.

Amb aquestes dades queda configurat l'Apache, en qualsevol moment podem editar-los al nostre gust o necessitats. Per a això hauríem d'editar el fitxer httpd.conf, que es troba en el directori d'instal·lació d'Apache, que hem indicat durant el procés d'instal·lació, més concretament, en el subdirectori conf.

Una vegada instal·lat, l'Apache es posa en funcionament, per a controlar l'Apache podem trobar en el botó d'inici, secció programes, un nou grup cridat "Apache HTTP Server", amb icones per a detenir el servidor, engegar-lo, editar l'arxiu de configuració httpd.conf o per a veure els logs d'accés. També podem veure en la barra de tasques una nova icona que indica que l'Apache està funcionant i que ofereix opcions per a controlar-lo si premem sobre ell.

D.3 COMPROVACIÓ DEL FUNCIONAMENT.

Per a comprovar que el servidor està actiu i funciona correctament podem obrir un explorador i provar a accedir a la URL <http://localhost>, que és el nostre propi servidor. Si tot ha anat bé observarem el missatge de benvinguda del servidor.



Figura 20. Apache funcionant

ANNEX E: INSTAL·LACIÓ DE PHP WINDOWS

E.1 INSTAL·LACIÓ DE PHP 5.0.1

Procedim a descarregar el PHP per a Windows l'arxiu està localitzat en la secció Downloads, apartat Windows Binaries, i haurem de localitzar el Zip Package (no el installer, encara que aquest ocupi menys després no ens servirà).

Una vegada descarregat tot el ZIP, fem una carpeta anomenada PHP en C:\PHP per instal·lar els arxius del servidor (PHP, MySQL...), després extraïem els arxius del ZIP a dintre d'aquesta carpeta PHP, tal que els continguts del ZIP quedaran dintre de la ruta C:\PHP\, ara cal agafar tots els arxius DLL localitzats en la carpeta principal C:\PHP\ i copiar-los al directori System32 (NT,2000,XP,2003) de la carpeta del Windows (els arxius DLL continguts en la carpeta EXT no fa falta copiar-los).

E.2 CONFIGURACIÓ DEL ARXIU PHP.INI

El següent pas és configurar el php.ini, copiem l'arxiu C:\PHP\php.ini-dist i li posem php.ini després ho editem amb el mateix bloc de notes, si ho necessitéssim editem la línia `register_globals = Off` i la col·loquem el valor: `register_globals = On`, activar aquesta directiva ens permet assumir que les variables són globals i poden arribar per qualsevol mètode (POST, GET, COOKIE, SERVER, etc).

Dintre del php.ini busquem `extension_dir` i li posem la carpeta que conté els arxius `phpxxx.dll`, que per defecte és la carpeta `ext` dintre de PHP. IMPORTANT!!: durant tota la configuració de directoris, hem d'utilitzar aquesta barra "/" i no aquesta "\", a més de posar-la entre cometes, o sigui, que ha de quedar així:

```
; Directory in which the loadable extensions (modules) reside.  
extension_dir = "C:/PHP/ext/"
```

A més podem activar les extensions que vulguem o necessitem, per a això busquem Windows Extensions i per a carregar les extensions traïem el ; de davant, per exemple, si volguéssim carregar l'extensió `gd2.dll` (per a manejar les funcions PHP relatives a imatges) hauríem de canviar

```
;extension=phpgd2.dll per  
extension=phpgd2.dll
```

Es pot veure per a què serveix cada extensió en el Manual oficial de PHP, com nota addicional, ressaltar que hi ha algunes extensions que requereixen de llibreries extra que no vénen en el paquet complet de PHP, i per a fer-les funcionar hauríem de

buscar aquestes llibreries. No haurem d'activar totes les llibreries perquè després sortiran errors que no es troba tal arxiu. Abans d'instal·lar una llibreria, haurem de comprovar que en la carpeta de PHP hi ha els arxius DLL corresponents.

Si anem a fer proves amb upload d'arxius via HTTP, hem d'indicar el directori on els arxius s'emmagatzemaran temporalment, per a això busquem `upload_tmp_dir` i li donem el valor d'una carpeta que existeixi. Per exemple, en el nostre directori `C:\PHP\` creem una carpeta `uploads`, pel que quedarà alguna cosa així:

```
; Temporary directory for HTTP uploaded files (will use system default if not
; specified).
upload_tmp_dir = "C:/PHP/uploads/"
```

Per a treballar amb sessions, hem d'especificar un directori on es guardin els arxius temporals, per fer això haurem de triar `session_save_path` i li donem el valor d'un directori que existeixi (o ho creguem): `session_save_path = "C:/PHP/sessions/"`, a continuació copiem el `php.ini` a la carpeta `C:\Windows\`.

E.3 `httpd.conf` PER APACHE 2.0.X

Dynamic Shared Object (DSO) Support és on es carreguen els mòduls. Aquí carregem el mòdul de PHP per a Apache, donant l'adreça de l'arxiu `php5apache2.dll` (PHP5) que ho conté, trobem fins a un exemple:

```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule php5_module C:/Servidor/PHP/php5apache2.dll
```

El directori predeterminat per a guardar les nostres pàgines és el `htdocs` de l'Apache: `C:\Arxius de programa\Apatxe Group\Apache\htdocs\` però podem canviar-lo fàcilment. vam buscar `DocumentRoot` i ens surt això:

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Archivos de programa/Apache Group/Apache/htdocs"
```

Substituïm la carpeta `htdocs` de l'Apache per la qual vulguem, tots els arxius que vagis a provar amb l'Apache haurien d'estar localitzats en aquesta carpeta. Per exemple ens

creem una carpeta WEB dintre del directori del servidor: DocumentRoot "C:/Servidor/WEB/" Pel que construirem tota la nostra pàgina dintre d'aquest directori, que equival al directori arrel del nostre host local.