

# **Comparativa entre Xindice i eXist com a sistemes de bases de dades XML natives**

**Joan Anton Pérez i Braña**

Estudiant ETIG

**Consultor: Oscar Celma Herrada**

Memoria Final 10/01/2005

## Dedicatòria i Agraïments

---

A M<sup>a</sup> Angels , la meva companya i mare de les meves filles, Estel i Ona

## Resum del TFC

---

Aquest treball és el resultat del procés d'aprenentatge que he realitzat durant el període comprès de setembre de 2004 fins a principis de gener de 2005. En ell he pretès descriure sintèticament els conceptes adquirits, començant pels aspectes bàsics del llenguatge i de les tecnologies lligades a XML, la utilització de XML per la representació i l'intercanvi de dades així com les característiques comunes que veiem en les XML:DB, és a dir les bases de dades centrades en documents amb format XML. He fet una descripció, de la qual es pot extreure una comparativa, entre el procés d'instal·lació, configuració i ús de dos exemples de sistemes XML:DB com són eXist i Xindice. Sobre aquests dos sistemes s'han realitzat proves per a l'administració de les bases de dades així com en l'ús de XPath i XQuery per a la realització de consultes. Per a finalitzar he desenvolupat una aplicació amb l'objectiu de posar a la pràctica els coneixements adquirits juntament amb l'aplicació els web services. El codi resultant de la present aplicació es pot consultar en els arxius adjunts a la present memòria.

## Índex de continguts

---

- Capítol 1.** [Introducció.](#)
- 1.1. *Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.*
  - 1.2. *Objectius del TFC.*
  - 1.3. *Enfocament i mètode seguit.*
  - 1.4. *Planificació del projecte.*
  - 1.5. *Productes obtinguts.*
  - 1.6. *Breu descripció dels altres capítols de la memòria.*
- Capítol 2.** [Breu introducció a XML](#)
- 2.1. *XML*
  - 2.2. *RDF*
  - 2.3. *FOAF*
- Capítol 3.** [Breu introducció a les BD:XML](#)
- 3.1. *Les BD:XML*
  - 3.2. *Introducció a eXist*
  - 3.3. *Introducció a Xindice*
- Capítol 4.** [Procés d'instal·lació d'eXist i Xindice](#)
- 4.1. *Instal·lació de Java a Linux*
  - 4.2. *Instal·lació de Tomcat*
  - 4.3. *Instal·lació d'eXist*
  - 4.4. *Instal·lació de Xindice*
- Capítol 5.** [Aplicació client per a eXist](#)
- 5.1. *Introducció*
  - 5.2. *Fent servir la shell*
  - 5.3. *Especificant Paràmetres*
  - 5.4. *Configuració General*
  - 5.5. *Emmagatzament de documents*
  - 5.6. *Esborrat de Col·leccions/Documents*
  - 5.7. *Execució de consultes*
  - 5.8. *Fent servir una Configuració de la Base de Dades alternativa*
  - 5.9. *Backup/Restore des de la línia de comandes*
- Capítol 6.** [Aplicació client per a Xindice](#)
- 6.1. *Administració de la Base de dades*
  - 6.2. *Manipulació de documents*
  - 6.3. *Administració del Servidor*
  - 6.4. *Consultes a la base de dades*
- Capítol 7.** [XPath](#)
- 7.1. *Característiques generals d'XPath*
  - 7.2. *Ús d'XPath a Xindice*
  - 7.3. *Ús d'XPath a eXist*
- Capítol 8.** [XQuery](#)
- 8.1. *Característiques generals de XQuery*
  - 8.2. *Estat Actual*
  - 8.3. *Extensions XQuery*
  - 8.4. *Lligam Java*
- Capítol 9.** [Els Serveis Web](#)
- 9.1. *Els serveis Web XML*
  - 9.2. *Els protocols SOAP i XLM-RPC*
  - 9.3. *UDDI i serveis web*
- Capítol 10.** [Exemple pràctic](#) Aplicació per a la gestió de dades d'un centre d'estudis
- 10.1. *Requisits previs.*
  - 10.2. *Anàlisi.*
  - 10.3. *Disseny.*

10.4. *Implementació.*

**Capítol 11.** Conclusions.

**Capítol 12.** Bibliografia.

# Capítol 1 Introducció

## 1.1 Justificació del TFC i context en el qual es desenvolupa: punt de partida i aportació del TFC.

El present TFC es basa en el desenvolupament d'un sistema informàtic que permeti realitzar cerques de persones dins d'una organització del tipus d'un Institut de Formació Professional en el qual s'imparteixen Cicles Formatius de Grau Mitjà, de Grau Superior i Batxillerat. La informació estarà desada en documents que incorporaran dades sobre els alumnes, els cursos, i la relació que hi ha entre aquests alumnes i diferents organitzacions ( per exemple empreses en les quals realitzen les seves pràctiques).

En el present TFC s'utilitzaran documents XML per estructurar i desar la informació esmentada., concretament l'estructura de descripció de recursos [RDF:FOAF](#) . FOAF és una iniciativa que permet descriure les persones dins l'àmbit de la web semàntica.

Els documents XML s'emmagatzemaran en una base de dades XML nativa (BD:XML).

Les consultes es realitzaran en els llenguatges XPath i XQuery.

S'utilitzarà SOAP per l'intercanvi d'informació en un ambient distribuït

Per últim la implementació es realitzarà en PHP i HTML.

Cal dir que la experiència prèvia en l'ús d'aquestes tecnologies es redueix a conceptes bàsics de HTML i en l'ús de sistemes gestors de bases de dades relacionals per tant m'ha suposat el descobriment de nous camps dins de les bases de dades i nous entorns i llenguatges en el desenvolupament d'aplicacions

## 1.2 Objectius del TFC.

Independentment de que l'objectiu d'aquest TFC és aprofundir en el coneixement del desenvolupament d'aplicacions basades en BD:XML basant-me en el desenvolupament d'un projecte pràctic, degut a que aquest es farà utilitzant algun d'aquests dos sistemes [eXist](#) i [Xindice](#). S'avaluaran les prestacions d'aquests dos sistemes fent especialment èmfasi en els avantatges i inconvenients d'ambdós.

S'avaluaran també els requisits de programari i maquinari per a dur a terme un projecte d'aquestes característiques i, si s'escau la seva valoració econòmica per a la seva posada en marxa.

## 1.3 Enfocament i mètode seguit.

Cal dir que el mètode seguit en el desenvolupament del present TFC ha estat el d'investigar tot el que tenia a veure en XML, degut al meu desconeixement de l'abast del treball i de les tecnologies lligades a XML. Posteriorment i a cada tasca realitzada he intentat tenir clar els aspectes generals per assolir amb èxit la següent tasca programada. En el procés de confecció de la present memoria he intentat de manera sintètica explicar els aspectes més importants de cada capítol i compementar amb enllaços per un aprofundiment en el tema tractat

## 1.4 Planificació del projecte.

id	Tasca	Data inici	Data fi	dies
1	Elements i atributs XML (estudi)	10/09/2004	15/09/2004	4d
2	Visió general contingut TFC	15/09/2004	22/09/2004	6d
3	Confecció Pla de Treball (PAC 1)	23/09/2004	27/09/2004	3d
4	DTD (visió general)	16/09/2004	22/09/2004	5d

5	Esquemes (estudi)	23/09/2004	27/09/2004	3d
6	RDF (estudi)	28/09/2004	04/10/2004	5d
7	FOAF (estudi)	04/10/2004	08/10/2004	5d
8	Configuració Xindice i eXists	08/10/2004	14/10/2004	5d
9	Creació de Documents RDF:FOAF	15/10/2004	21/10/2004	5d
10	XSL (estudi)	21/10/2004	27/10/2004	5d
11	Confecció de plantilles XSL	26/10/2004	01/11/2004	5d
12	Confecció PAC 2	28/10/2004	02/11/2004	4d
13	XPath (estudi)	03/11/2004	05/11/2004	3d
14	XQuery (estudi)	08/11/2004	11/11/2004	4d
15	Creació de consultes( part pràctica)	12/11/2004	30/11/2004	18d
16	Revaluació de la PAC2	24/11/2004	26/11/2004	3d
17	Confecció PAC 3	02/12/2004	09/12/2004	6d
18	Implementació de l'aplicació	10/12/2004	02/01/2005	18d
19	Confecció Memòria + Presentació	03/01/2005	10/01/2005	6d

## 1.5 Productes obtinguts.

S'ha desenvolupat una aplicació per a la gestió d'un centre d'estudis que funcioni a internet emprant browser com a client i un servidor web Apache, on s'ubica l'aplicació, de manera que aquest últim realitza consultes a un servidor Tomcat , on s'ubicaria eXist com a sistema XML:DB. Aquestes consultes es realitzarien mitjançant Soap i WSDL i demostrar la comprensió final del que he exposat en el present treball realitzant consultes XQuery hostatjades en codi PHP per a generar codi HTML i poder ser visualitzat i permetent la interacció amb l'usuari

## 1.6 Breu descripció dels altres capítols de la memòria.

*En el capítol 2 es fa una breu introducció a la especificació XML i al conjunt de tecnologies que s'han desenvolupat al voltant d'aquest llenguatge*

*En el capítol 3 ens introduïm en el camp de les XML:DB veient les característiques generals i comparant aquestes amb les bases de dades relacionals. Així mateix es fa un comentari de les característiques de dos sistemes XML:DB com són eXist i Xindice.*

*En el capítol 4 es descriuen els requeriments necessaris per la instal·lació d'eXist i Xindice*

*En els capítols 5 i 6 fem una descripció de les característiques i possibilitats d'ús d'eXist i Xindice.*

*En el capítol 7 i 8 es veuen XPath i Xquery com a llenguatges de consulta i es donen exemples usant tant eXist com Xindice*

*En el capítol 9 es fa una introducció als serveis web i la seva repercussió en el desenvolupament d'aplicacions que funcionen a internet*

*En el capítol 10 es descriu el procés de desenvolupament del producte final que serveix com a exemple a la present pràctica*

*En el capítol 11 hi trobem les conclusions finals*

# Capítol 2 Breu introducció a XML

## 2.1 Característiques de XML

---

### 2.1.1 Especificació XML

XML és un estàndard internacional desenvolupat per un Grup de treball XML, conegut com el Comitè de Revisió Editorial de SGML, i format sota l'auspici del World Wide Web Consortium (W3C) a 1996. Aquest Treball es resumeix en la Recomanació de XML disponible a <http://www.w3.org/TR/REC-xml/>. També es interessant visitar la pàgina central de XML a W3C: <http://www.w3.org/XML> que conté força informació sobre XML.

La Recomanació diu que és "El Llenguatge Extensible de Marques abreujat XML, i aquesta descriu una classe d'objectes de dades anomenats documents XML i parcialment descriu el comportament de programes d'ordinador que poden processar-los".

XML s'ha pensat per a treballar a la Web, i una possible conseqüència és que pot utilitzar-se amb mecanismes i protocols que ja funcionen com HTTP, MIME, les URL. No necessita cap requeriment afegit.

Les regles de marcat de XML es basen en un llenguatge de marcat anterior denominat llenguatge de marcatge generalitzat estàndard [SGML](#) com també ho és HTML.

Qualsevol document conforme amb XML és conforme amb SGML, fins i tot la majoria de les aplicacions de SGML poden convertir-se en XML, aquesta propietat condiona alguns aspectes de XML i fan que inclogui certes restriccions

### 2.1.2 Diferències entre XML i HTML

HTML està orientat a la representació de dades en canvi XML està orientat a les dades en si mateixes. Mentre HTML defineix un conjunt d'etiquetes i atributs vàlids, una utilització vàlida d'aquests elements i un significat visual per a cada element, XML no defineix etiquetes ni com s'han d'utilitzar si no que defineix un conjunt de regles sintàctiques bàsiques per a crear documents. És per això que XML és un metallenguatge, és a dir, un llenguatge per a definir altres llenguatges. Un no substitueix a l'altre ja que s'utilitzen per a tasques diferents, mentre HTML s'utilitza per a presentar informació, XML es fa servir per a representar i intercanviar dades, independentment de la seva presentació. És per això que es poden considerar complementaris.

### 2.1.3 XML com a conjunt de tecnologies

XML és en realitat un conjunt d'estàndards, tots relacionats i complementaris i que són els següents:

- [CSS](#) el llenguatge de fulls d'estil que es pot aplicar a XML igual que a HTML
- [XSL](#), eXtensible Style Language. Afegeix a un document XML una descripció de com cal visualitzar-se i que inclou certa capacitat de procés i transformació d'aquests documents.
- [XML Namespaces](#). Els espais de noms proveeixen d'un context al que s'apliquen les marques d'un document XML i que les diferencia d'altres amb el mateix nom però que són vàlides en altres contextos.
- [XML Schemas 1 i 2](#) Els esquemes son una manera de definir restriccions que seran aplicades al document XML. El terme Schemas s'ha pres de mon de les bases de dades relacionals, on un esquema es refereix al model al que un conjunt de dades s'ajusten. XML proveeix d'una manera de definir aquests esquemes.
- [RDF](#), l'estàndard del W3C per a metadades, s'utilitza per enllaçar cada metadada a un arxiu definint el tipus d'aquesta metadada.
- [XPath](#), s'ha dissenyat com a medi per a fer referències a porcions específiques d'un document XML i per tant ha estat la base per a poder fer consultes en un document o en una col·lecció de documents.
- [XLink](#) que descriu una manera estàndard per afegir hiper-enllaços a un arxiu XML.



- [XPointer](#) XPointer s'utilitza per identificar l'estructura interna d'un document XML seleccionant parts específiques del document en funció de la informació de l'arbre del document.
- [XQuery](#) que podríem considerar com una extensió de XPath permet realitzar consultes d'una manera més eficient donant més recursos pel desenvolupador.
- [XUpdate](#) la millora sobre XPath i XQuery és que permet les actualitzacions en documents XML.
- [DOM](#) Sèrie de funcions estàndard emprades per a manipular arxius XML i HTML des de un llenguatge de programació.

Degut a aquesta diversitat i al seu continu creixement cal consultar regularment [pàgina d'informes tècnics del W3C](#)

### 2.1.4 Aportacions de XML

Breument podem dir que els avantatges d'usar XML són:

- Independència de les dades respecte de les aplicacions. D'acord amb el principi de "Gestió d'Informació Oberta". La seva normalització internacional permet que sigui accessible a aquell que ho desitgi, no depèn de cap fabricant i no es necessari pagar per utilitzar-lo.
- Informació sobre informació, un context força útil en el seu processament informàtic. Degut a que un document es compon de tres parts: semàntica, estructura i presentació les etiquetes XML s'utilitzaran per marcar la semàntica i l'estructura del document.
- Elements per a descriure l'estructura d'un document.
- Eines per a organitzar la informació segons el nostre desig permetent emmagatzemar els documents en diferents fitxers. Permet integració de diferents formats, gràfics de vídeo de so dins d'un mateix document
- Formes de garantir que les dades compleixen amb certes restriccions
- Mètodes per a visualitzar la informació
- Capacitat per a passar informació a les aplicacions
- Enllaços per a les dades relacionades.

## 2.2 RDF

XML utilitza un sistema d'identificació de la informació continguda en un document o aplicació de XML. Aquest sistema s'anomena estructura de descripció de recursos. (RDF, Resource Description Framework).

RDF és una infraestructura que permet la codificació, intercanvi i reutilització per varis documents de metadades estructurats. RDF és una aplicació de XML. Permet als autors dels documents publicar vocabularis llegibles pels humans i processables per màquines. El W3C va adoptar la seva especificació en febrer de 1999 . La especificació actual és pot consultar a <http://www.w3.org/RDF/> . La estructura rígida que imposa RDF suporta un sistema de metadades estandarditzat, facilitant així l'intercanvi de metadades definits per diferents comunitats de recursos.

RDF és un sistema en el que es descriu un recurs per mitjà de les seves propietats relacionades. Cada descripció de propietat es compon d'una sentència que a la vegada aquesta es compon d'un tipus de propietat, un valor i un identificador de recurs sobre el que s'aplica la propietat.

L'espai de noms de RDF és:

```
<RDF xmlns="http://www.w3.org/TR/REC-rdf-syntax#" >
```

Una descripció RDF es compon recursos, propietats i sentències. Un recurs pot ser una pàgina web, fragments d'aquesta, un element específic de HTML o XML, una base de dades o una imatge que pot o no estar associada a una pàgina web. Tots els recursos disposen d'un identificador de recursos, URI, Uniform Resource Identifier i un identificació de connexió opcional que es pot emprar per descriure'l. Una propietat és una característica, atribut o relació específica que es pot utilitzar per a descriure un recurs. Quan s'uneix una propietat i el seu valor per a un recurs neix una sentència RDF. Les sentències tenen tres parts: subjecte,

predicat i objecte. El subjecte és l'element a que fa referència la sentència. El predicat és el paper que fa el subjecte en relació al recurs i l'objecte és el valor de la propietat.

D'aquesta manera si tenim una sentència llegible per una persona:  
Joan Anton és el desenvolupador de <http://www.elmeuservidor.com>

Part de la sentència	Valor
Subjecte(Recurs):	<a href="http://www.elmeuservidor.com">http://www.elmeuservidor.com</a>
Predicat (Propietat)	desenvolupador
Objecte(Literal)	Joan Anton

De manera que la sentència seria:

```
<RDF xmlns="http://www.w3.org/TR/REC-rdf-syntax#">
< Desenvolupador about="http://www.elmeuservidor.com">
  Joan Anton
</Desenvolupador>
</RDF>
```

Quan s'utilitza RDF per a descriure una col·lecció de recursos és necessari utilitzar un contenidor. RDF defineix tres tipus de contenidors diferents: *bag* (bossa), *sequence* (seqüència) i *alternative* (alternativa).

Els contenidors de tipus bag permeten declarar una propietat que conté més d'un valor de manera que l'ordre de processament d'aquest no sigui important. Aquests contenidors permeten duplicats. Els contenidors de tipus sequence en canvi s'utilitzen per remarcar un ordre específic. També permeten valors duplicats. Els contenidors de tipus alternative proporcionen una llista de recursos o literals que representen alternatives per a un únic valor de la propietat.

## 2.3 RDF:FOAF

La iniciativa [Friend of a Friend](#) (FOAF), no té aspiracions comercials, la seva organització es horitzontal y proposa descriure *relacions* en llenguatge llegible pels ordinadors per a crear xarxes de relacions que existeixen entre amics i de certes propietats de la persona que s'està descrivint.

FOAF és una aplicació basada en XML, RDF y OWL ([llenguatge d'ontologies web](#)), tecnologies recomanades pel W3C. [Leandro Mariano \(inkel\)](#) explica que es tracta d' un arxiu XML (en realitat, RDF escrit en XML) del que es descriuen persones, documents o qualsevol cosa. En aquest arxiu que ara anomenarem FOAF, indiquem a quines persones coneixem, quins són els meus amics i qui no, quins són els meus projectes actuals i anteriors, quin grau d'interès tinc per a uns temes o uns altres, és a dir que es poden definir molts trets.

Per a conèixer més a fons l'especificació de FOAF es pot consultar a:

<http://xmlns.com/foaf/0.1/>

Per a Dan Brickley (danbri), un dels cofundadors del projecte FOAF juntament amb Libby Miller (libby), la idea és molt senzilla, un arxiu que pugui ser interpretat per un ordinador, tecnologia que aprofundeix en el camp de la web semàntica. Una vegada disposem d'aquests arxius podem navegar per tots els documents llegibles que es trobin al web i disposar d'una base immensa per extraure la informació que necessitem per a crear la base de dades a la nostra mida. Possiblement podrien conèixer dades vàlides pel que fa a la seva veracitat amb d'altres que realment siguin mitges veritats i fins i tot mentides.

En aquests moments ja hi ha un munt d'aplicacions per a conèixer l'experiència de la web semàntica. Gràcies a [foaf-a-matic](#), disposem d'una pàgina web en la que mitjançant javascript podem aconseguir crear el propi arxiu FOAF que posteriorment caldrà col·locar en el nostre espai web per a que comenci a propagar-se. A [FoafSpace](#) es pot navegar per tots els arxius FOAF; Morten Frederiksen ha creat un [explorador de FOAF](#) i Jim Ley [FOAFNAUT](#). [FOAFboat](#) ofereix informació FOAF dels canals IRC; [Codepiction](#) de Danbri i Libby, és una eina que ofereix informació sobre imatges....

Una de les aplicacions més interessants i populars que he trobat és una una representació en [SVG](#) (gràfics vectorials, basat en XML i apte per a la Web) de les relacions que existeixen entre les 100 empreses més grans de EE.UU., creada per Josh On de Future Farmers per a They Rule i titulada [foafCORP](#). En general, la comunitat més activa de FOAF es reuneix entorn del canal #foaf de irc.freenode.net. En aquests moments podem trobar més de dos milions de descripcions FOAF en el mon (a la Web), i la xifra continua creixent. De tot això i sobre la potència de FOAF, Jim Ley (jibberjim) afirma que la diferència entre FOAF i altres projectes similars es que *FOAF és el vocabulari que permet construir sistemes de xarxes socials i aquestes dades estan on cadascú vol que estiguin.*

# Capítol 3 Breu Introducció a les BD:XML

## 3.1. Aspectes generals de les BD:XML

### 3.1.1 Introducció

És conegut per tothom que 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 el futur de l'administració de dades ja que ha permès trencar certes barreres i crear un format estàndard per a processar informació.

Doncs bé, XML està provocant l'aparició de noves tecnologies, entre elles l'aparició d'una nova generació de bases de dades que encara que ara per ara es troben en una fase d'investigació i desenvolupament, 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)..

### 3.1.2 Tractament XML en les bases de dades relacionals

Aquests tipus de bases de dades són completament diferents de les relacionals, les quals en aquests moments ja donen suport per a XML, però encara segueixen emmagatzemant la informació de manera relacional, és a dir en forma tabular o emmagatzemen tot el document en format Binary Large Object. A més no estan ben preparades per a emmagatzemar estructures de tipus jeràrquic com són els documents XML per les següents raons:

- Les BBDD relacionals tenen una estructura regular mentre que els documents XML tenen caràcter heterogeni.
- Els documents XML solen contenir molts nivells d'aniuament mentre que les dades relacionals són "planes" és a dir tenen un nombre de columnes per a cada fila
- Els documents XML tenen un ordre intrínsec mentre que les dades relacionals són no ordenats
- Les dades relacionals són generalment denses , és a dir cada columna té un valor i acostumen a tenir pocs valors nuls, en canvi els documents XML són dispersos ja que poden presentar mancança d'informació mitjançant l'absència de l'element

S'han creat teories per a encaixar objectes o estructures jerarquitzades en bases de dades relacionals. Per exemple les regles de Transformació relacional en DTD

- *Regla 1* • Per a cada taula en l'esquema de la base de dades cal crear un element amb el mateix nom de la taula i la cardinalitat apropiada
- *Regla 2* • Les columnes de la taula són incloses en un altre element, és a dir un subelement de l'element creat mitjançant la regla anterior, que representa un registre de la taula
- *Regla 3* • Per a cada columna de la taula que la seva dada és simple, (char, integer, etc.) cal crear un element, sub-element de l'element creat en el pas anterior, de tipus
- *Regla 4* • Per a cada columna de la taula en el que el seu tipus de dades és complex, és a dir de tipus objecte, caldrà crear un element complex , sub-element de l'element creat en el pas 2, amb el mateix nom de la columna. Per a cada propietat del tipus objecte caldrà crear un element amb el mateix nom de la propietat
- *Regla 5* Per a cada columna en la taula que és una taula aniuada, crear un element amb el mateix nom d'aquella columna i la cardinalitat adequada. Caldrà també repetir els passos des de el 2 .

### 3.1.3 Comparativa entre les XML:DB i les bases de dades relacionals

La principal característica de les XML:DB és la capacitat d'obtenir resultats de consultes amb format XML i es per això que aquestes bases de dades pertanyen a la categoria "XML-enabled database"

L'organització "XML:DB Initiative for XML Databases" descriu una base de dades d'aquest tipus com un: "model lògic" per a documents XML i emmagatzema i recupera documents d'acord amb aquest model.

Totes les bases de dades relacionals són centrades en les Dades, “*data-centric databases*”, ja que el que elles emmagatzemen són dades atòmiques. Una BD:XML, ni té camps, ni emmagatzema dades atòmiques ja que el que ella emmagatzema són documents XML, per la qual cosa a aquest tipus de bases de dades se les anomena bases de dades centrades en documents, “*document-centric databases*”.

### 3.1.4 Característiques generals de les XML:DB

Diversos productes ofereixen diferents característiques per a les bases de dades natives XML, i són generalment aquestes:

Són bases de dades i suporten transaccions, accés multiusuari, llenguatges de consulta, etc dissenyats especialment per emmagatzemar documents XML

#### Emmagatzemament

Per deducció lògica una base de dades nativa en XML emmagatzema informació en format XML però això no és solament una deducció lògica ja que aquest tipus de bases de dades té repositoris amb un format tipus XML com pot ser DOM o Infoset. En aquest mateix repositori s'emmagatzemen els índexs que es generen per a cada document XML emmagatzemat. És a dir emmagatzema el document XML sencer en format text i proporciona alguna funcionalitat a la base de dades per accedir a ell

- Possibilitat senzilla: Emmagatzemar el document com un BLOB en una base de dades relacional o mitjançant un fitxer i proporcionar certs índexs sobre el document que accelerin l'accés a la informació
- Possibilitat sofisticada: emmagatzemar el document en un magatzem adequat amb subíndexos, suport per a transaccions, etc.

Emmagatzemament basat en el model: Magatzema un model binari del document (p. e. DOM) en un magatzem existent o en un específic

- Possibilitat 1: Traduir el DOM a taules relacionals com Elements, Atributs, Entitats, etc.
- Possibilitat 2: Traduir el DOM a objectes en una BDOO
- Possibilitat 3: Utilitzar un magatzem creat específicament per aquets fi

#### Emmagatzemament de documents en col·leccions:

Les col·leccions juguen en les bases de dades natives el paper de les taules en les bases de dades relacionals. Els documents s'agupen en funció de la informació que contenen en col·leccions que a la vegada poden contenir altres subcol·leccions dins d'elles.

#### Validació de documents:

Cal que els documents que s'emmagatzemin siguin “XML ben formats”

#### Processament de dades

El processament de dades en aquest tipus de bases de dades sembla ser molt àgil però això no és del tot així degut al format jeràrquic en el que està emmagatzemada la informació. Moltes bases de dades necessiten que es recuperi tot el document XML, s'actualitzi en la API XML que s'estigui fent servir i posteriorment es torni a emmagatzemar en el repositori. Això és degut a que no existeix un llenguatge estàndard que permeti l'actualització, inserció o eliminació d'elements d'un document XML. Existeix un llenguatge que permet realitzar actualitzacions en un document XML encara que alguns gestors d'aquest tipus de bases de dades no el suporten, aquest llenguatge és XUpdate (<http://www.xmldb.org/xupdate>)

#### Consultes

Aquest tipus de bases de dades no utilitzen SQL com a llenguatge de consulta. En lloc d'aquest utilitzen XPath o XQuery. Algunes bases de dades permeten seleccionar els elements que deuran tenir índexos mentre que d'altres indexen tot el contingut del document. El problema que tenen les bases de dades que només permeten cerques amb XPath és que no permeten realitzar cerques gaire complicades ja que no permeten l'ordenació ni el cross join ja que

inicialment XPath va ser creat inicialment per a realitzar cerques dins d'un document i no en una base de dades.

No obstant cal dir que tot això està en continua evolució. Ara la majoria de les BD:XML suporten un o més llenguatges de consulta. Un dels més populars és XQuery.

### Indexació XML

S'ha de permetre la creació d'índexos que accelerin les consultes realitzades freqüentment

#### Creació d'identificadors únics

A cada document XML se li associa un identificador únic pel que serà reconegut dins del repositori

## 3.2. Introducció a eXist

---

eXist és el que [XML:DB Initiative](#) anomena com un Native XML Database. Està basat en [XQuery](#), indexació automàtica, extensions per a cerca de text complet, suporta [XUpdate](#) i integració estreta amb les eines de desenvolupament XML més exteses.

La implementació actual de les eines de base de dades és XQuery 1.0 treballat com a esborrany de novembre, 2003, amb la excepció de les característiques relacionades amb l'XML schema .

El nou XQuery suportat en eXist fa possible l'escriptura d'aplicacions web senceres fetes amb XQuery y XSLT. Els arxius XQuery poden ser directament passats a la base de dades , usant el XQueryServlet, el XQueryGenerator per a Cocoon o usant l'estil-REST d'API (veure [Guia del Desenvolupador](#)).

Les característiques implementades inclouen: definició d'usuaris i mòduls, *checking* de tipus estàtic, lligam amb Java, funcions de cerca de text, mòduls d'extensió per a suport HTTP, manipulació de base de dades i transformacions basades en XSL

Per a evitar l'ús de memòria intensiu a causa del document en arbre transversal mentre dura el processament de la consulta, eXist usa una estructura d'index eficaç que està basada en un esquema d'indexació numèrica per l'identificació de nodes XML en l'índex . La base de dades lleugera, completament escrita en Java i pot ser fàcilment desplegada en un gran nombre de maneres , executant-se ambdues com un únic procés de servidor, dins un servlet-motor o directament empoltrat dins una aplicació. Per a un resum de les característiques d' eXist si us plau cal mirar el [full de característiques](#) .

eXist ofereix una línia de comandes integrada a la GUI gràcies a la classe [org.exist.client.InteractiveClient](#). Està basada en l'API de XML:DB i proporciona al client totes les comandes per a la majoria de les tasques relacionades amb la base de dades com crear o eliminar col·leccions, gestió d'usuaris backup/restore XML i consultes.

## 3.3. Introducció a Xindice

---

### 3.3.1 Què és Xindice

El servidor Xindice Core és un servidor de bases de dades dissenyat per emmagatzemar dades XML. El servidor Xindice és el que [XML:DB Initiative](#) anomena com un Native XML Database..

Totes les dades que entren i surten del servidor són XML. El llenguatge de consulta que es fa servir és XPath i suporta les APIs DOM i SAX. Mentres treballem amb dades XML i Xindice senzillament dissenyem les dades com XML i les emmagatzemem com XML.

Això proporciona un mecanisme extremadament flexible per modelar les dades i fins i tot ens permetrà modelar construccions que són més dificultoses mitjançant mecanismes tradicionals. Aquest és un model semiestructurat i això per alguns casos és un component essencial. Fent servir una base de dades nativa de XML com Xindice per emmagatzemar aquestes dades ens podem centrar en la construcció dels seus usos i no preocupar-nos dels esquemes complexos de la seva construcció com per exemple de les restriccions.

En canvi, si les dades no són de tipus XML o es necessita un control exacte sobre l'estructura de les dades llavors pot ser millor usar un altre solució per la base de dades.

### 3.3.2 Estat actual

El servidor Xindice suporta l'emmagatzament de documents XML ben formats. No hi ha cap esquema preconcebut per a determinar com s'han d'ubicar els documents dins de les col·leccions. Això fa que Xindice sigui una base de dades semiestructurada la qual cosa permet molta flexibilitat en com s'han d'emmagatzemar les dades. En l'estat actual Xindice ja es pot dir que és una poderosa eina per emmagatzemar dades XML, encara que hi ha molt per fer.

### 3.3.3 Resum de les característiques

**Col·leccions de Documents:** Els documents són emmagatzemats en col·leccions les quals poden ser consultades en la seva totalitat. Es poden crear col·leccions que continguin tant sols els documents del mateix tipus o podem col·locar-los tots junts dins d'una mateixa col·lecció.

**Motor de Consultes XPath:** Per a consultar les col·leccions de documents es pot usar Xpath de la mateixa manera com està definit a W3C. Això proporciona un mecanisme bastant flexible per a consultar documents navegant al seu través i restringint l'arbre de resultat que sigui retornat.

**Indexació XML:** Per a millorar el rendiment de les consultes sobre gran quantitat de documents, podem afegir índexos sobre elements o valors d'atributs. Això pot accelerar el temps de resposta a la consulta.

**Implementació de XUpdate:** Mentre s'emmagatzema el document XML a la base de dades podem realitzar canvis sense recuperar el document sencer. El mecanisme que permet això es fa mitjançant XUpdate. Aquest és un llenguatge basat per a fer modificacions específiques tant a col·leccions senceres com a documents concrets.

**Implementació de l'API Java XML:DB:** Pels programadors de Java, Xindice proveeix una implementació de l'API XML:DB. Aquesta API està pensada per dotar de certa portabilitat a les aplicacions basades en bases de dades XML de la mateixa manera que ho fa JDBC amb les bases de dades relacionals.

**Gestió de les eines de la línia de comandes:** Per ajudar a l'administrador, Xindice proveeix d'un conjunt de comandes de gestió de manera que tot allò que es pot fer des de l'API XML:DB es pot fer també des de la línia de comandes.

**Arquitectura Modular:** El servidor Xindice està construït d'una manera molt modular. Això simplifica l'afegir i treure components per adaptar el servidor a un ambient particular o adaptar-lo a un altre ús.

### 3.3.4 Estructura de la Base de dades

El servidor Xindice està dissenyat per emmagatzemar col·leccions de documents XML. Les col·leccions poden ser estructurades com una jerarquia similar a la d'un sistema de fitxers típic de Windows o Unix.

A Xindice el magatzem de dades és arrelat com una instància d'una base de dades que pot ser també usada com una col·lecció de documents. Aquesta base de dades pot contenir un cert nombre de col·leccions. En una instal·lació per defecte la instància de la base de dades és anomenada "db" i tots els paths corresponents a les col·leccions neixen de /db. És possible reanomenar la instància de la base de dades però no és gens necessari.

### 3.3.5 Introducció a la línia de comandes

El servidor Xindice un programa anomenat xindice que permet gestionar les dades emmagatzemades en el servidor. Una llista completa de les comandes disponibles i forces detalls de cadascuna la trobem a [Command Line Tools Reference Guide](#).

L'eina *xindice* està localitzada en el directori Xindice-Core/bin i és una bona idea afegir aquest directori en la nostra variable d'entorn PATH. Tots els exemples d'aquest treball asumiran que Xindice-Core/bin és en el PATH del sistema operatiu.

## Capítol 4 Procés d'instal·lació d'eXist i Xindice

Tot el que s'explica a continuació correspon al procés de configuració realitzat sobre un sistema operatiu Linux Red Hat 9.0.

### 4.1 Instal·lació de Java

#### 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.

## Comprovacions.

Per a comprovar si Java ha estat instal·lat correctament, s'executa el següent des de el terminal

```
which java
```

Lo 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.

## 4.2 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/startup.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](#).

### Comprovar si el servidor està engegat

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.

### 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.

## 4.3 Instal·lació de Tomcat

### Instal·lació de Tomcat

1. Cal baixar la versió **binària** de Tomcat a: <http://jakarta.apache.org/tomcat> . En aquest treball s'utilitza el binari comprimit *jakarta-tomcat-5.0.28.tar.gz*
2. Cal descomprimir l'arxiu .tar de Tomcat a */usr/local/*, això genera un directori anomenat *jakarta-tomcat-5.0.28*, per a donar uniformitat es canvia el nom per *tomcat*
3. Es defineix la variable d'ambient que indicarà al sistema on es troba Tomcat, aquesta variable s'anomena *CATALINA\_HOME* i cal ser afegida a l'arxiu */etc/bashrc*, a Linux es fa afegint la línia: *export CATALINA\_HOME=/usr/local/tomcat;*

### Configuració Local

En aquest instant i per a fer posteriors proves el que es fa es modificar l'arxiu */etc/hosts* afegint la línia:

```
127.0.0.1    www.elmeuservidor.com
```

Caldria fer-ho mitjançant DNS però per alleugerir el treball s'ha optat per modificar */etc/hosts*.

### Execució i Proves

Un cop efectuades les fases anteriors es pot realitzar un joc de proves inicial sobre Tomcat. Per a fer això cal executar el que ara es descriu :

1. Anar al directori bin de Tomcat (*/usr/local/tomcat/bin* ) i executar l'arxiu *startup.sh* o també *./catalina.sh run*:

```
[root@servidor1 bin]# ./catalina.sh run
Using CATALINA_BASE: /usr/local/tomcat
Using CATALINA_HOME: /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JAVA_HOME: /usr/local/j2sdk1.4.2_06/
```

2. Encara que la pantalla es pugui parar mitjançant Crtl+C i continuar realitzant tasques dins d'ella, el servidor s'executarà fins que això s'aturi mitjançant l'execució de *shutdown.sh* (o també *./catalina.sh stop* ), fins i tot si intentem executar *./startup.sh* consecutivament, sense executar *shutdown.sh* es genera un error..
3. Assegurant-nos que Tomcat estigui actiu, des del navegador del sistema s'ha de poder visitar l'adreça que va ser definida a */etc/hosts*, en el meu exemple com *www.elmeuservidor.com* afegint el fragment *:8080* al final:

```
www.elmeuservidor.com:8080
```

A l'introduir l'adreça anterior en un navegador ens indica que sol·liciti la pàgina principal [www.elmeusevidor.com](http://www.elmeusevidor.com) en el port TCP 8080; [www.elmeusevidor.com](http://www.elmeusevidor.com) està a la màquina local (127.0.0.1) tal com s'ha definit a l'arxiu `/etc/hosts` i el port 8080 és on precisament està (s'indica en la seqüència de `startup.sh`)

4. En el navegador apareix la pàgina principal de la documentació de Tomcat i els enllaços cap als diferents exemples de Tomcat.

## 4.4 Instal·lar eXist com una Aplicació Web

Per a instal·lar eXist amb servlet-motor existent, en el nostre cas Tomcat, cal descarregar l'arxiu `.war` i ubicar-lo dins la localització adient on el servlet-motor el pugui trobar. La majoria de les instal·lacions tenen un directori anomenat `webapps` i automàticament el carregarà com aplicació web el fitxer `.war` que trobi enllà. Cal tornar a nomenar l'arxiu a `exist.war` (si no alguns exemples no poden treballar correctament).

### **Exemple : desempaquetar l'arxiu .war**

```
mkdir exist
cd exist
jar xfv ../exist.war
```

Finalment s'elimina l'arxiu `.war`

Es pot reiniciar el servlet-motor de nou. Si eXist està executant-se dins d'un servlet-motor, el directori d'inici sempre estarà en `WEB-INF`.

Aquí és on hi ha el principal fitxer de configuració `conf.xml` i on resideixen els fitxers de bases de dades. Els fitxers de log es troben a `WEB-INF/log`

## 4.5 Instal·lació de Xindice

### Prerrequisits abans de la Compilació

Cal tenir instal·lada una versió superior del JDK 1.3, en el cas que ens ocupa s'ha instal·lat el `j2sdk1.4.2_06` segons les passes indicades en el punt [4.1](#).

Cal tenir instal·lat el servidor web Tomcat, descrit en el punt [4.3](#)

Cal descarregar els arxius font de Xindice des de la ubicació <http://xml.apache.org/xindice/download.cgi>. En el cas que ens ocupa s'ha descarregat l'arxiu comprimit per a Linux de la versió 1.1b4 `xml-xindice-1.1b4-src.tar.gz`.

### Compilació de Xindice

Es desempaqueta l'arxiu `xml-xindice-1.1b4-src.tar.gz` de manera que es crea el directori `xindice-1.1b4`. Per homogeneïtat amb la feina realitzada fins aquest moment es mou aquest directori a la ubicació `/usr/local` i canviem el nom del directori pel de `xindice`. Ja dins del directori `/usr/local/xindice/` executem l'arxiu `build.sh` de manera que ens apareix un missatge com aquest:

```
[root@localhost xindice]# ./build.sh
```

```
Apache Xindice Build System
```

```
-----
Buildfile:build.xml
```

```
BUILD SUCCESSFUL
```

```
Total time 13 seconds
```

Per a comprobar que el procés ha estat un èxit verifiquem el contingut del directori `usr/local/xindice/dist`

```
[root@localhost xindice]# ls -l dist
total 4692
-rw-r--r--      1 root   root      464593 nov  2 21:38 xindice-1.1b3.jar
-rw-r--r--      1 root   root    4313586 nov  2 21:38 xindice-1.1b3.war
-rw-r--r--      1 root   root       248 nov  2 21:38 xindice-1.1b3.xml
```

## Instal·lació de Xindice amb Tomcat

Partint de la versió 1.1, de Xindice aquest no és un servidor independent més. Les funcions del servidor ara es basen en un ús obeient del servlet 2.2 (o 2.3) preferit. S'ha provat i s'ha demostrat el treball de Xindice sota Tomcat i Jetty i sembla ser que ara per ara no hi ha cap raó per a esperar un mal funcionament sota l'ús d'altres servidors.

La instal·lació llavors és directa: tant sols cal desplegar l'arxiu.war de Xindice (xindice-1.1b4.war) en el servidor preferit i ja està llest. Tant sols cal tenir dos punts en compte:

- El punt final de Xindice XML-RPC és configurat en el client com `http://algunservidor:algunport/xindice/`. Això significa que és molt recomanable desplegar l'arxiu .war de Xindice sota el contexte de xindice. Això és pot aconseguir fàcilment sota Tomcat renomenant l'arxiu.war com a xindice.war o (en Tomcat 4.1.x) copiant l'arxiu `dist/xindice-1.1b4.xml` sota el directori de `$CATALINA_HOME/webapps`. Cal tenir en compte que en algunes versions de Tomcat caldrà engegar dues vegades el servidor per primera vegada per a que pugui configurar-se correctament.
- És possible que sigui necessari corregir l'arxiu de la configuració de Xindice que resideix sota `/Web-INF/system.xml`. Aquest arxiu configura entre altres coses la localització física de la base de dades. Per defecte, les dades estan per sota de [ localització de l'empaquetat war]/`Web-INF/db`, però no pot ser una bona idea per a molts usuaris: deixaria la base de dades com perduda si una actualització succeís de manera inadvertida ja que el directori seria sobreescrit. També cal dir que si el servidor no està desempaquetant l'arxiu .war llavors Xindice no podrà començar.

Tenir el servidor empaquetat dins webapp significa també que engegar i aturar Xindice és justament una qüestió de parar/engegar el servidor en ús.

Copiem, utilitzant el mateix raonament que s'ha fet servir per la configuració d'eXist amb Tomcat, l'arxiu .war al directori webapps que trobem dins de `$CATALINA_HOME`.

```
[root@localhost xindice]# cp dist/xindice-1.1b4.war $CATALINA_HOME/webapps/xindice.war
```

Desempaquetem l'arxiu *xindice.war*

```
jar xfv ../xindice.war
```

Això fa que es crei el directori xindice on trobem dins els directoris *META-INF* i *WEB-INF*

És dins de `$CATALINA_HOME/webapps/xindice/WEB-INF/db` on s'emmagatzema la base de dades per defecte.

Ara ja està tot configurat arrencar Xindice dins de Tomcat. Per això hem d'anar al directori `$CATALINA_HOME/bin` i executar l'arxiu *startup.sh* que arrenca el servidor Tomcat (fa el mateix que `./catalina.sh run` però aquest script a diferència de `./catalina.sh` finalitza la seva execució ja que sembla que executa `./catalina.sh run` en segon pla ). La resposta que veiem des del terminal és la següent

```
[root@localhost xindice]# cd $CATALINA_HOME/bin
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat
Using CATALINA_HOME:   /usr/local/tomcat
Using CATALINA_TMPDIR: /usr/local/tomcat/temp
Using JAVA_HOME:       /usr/ljava/j2sdk1.4.2_06/
[root@localhost xindice]#
```

Per a finalitzar podem comprovar que Xindice està corrent si mitjançant mozilla i a l' URL `http://localhost:8080/xindice` o també degut a la configuració realitzada abans `http://www.elmeuservidor.com:8080/xindice` , veiem el el contingut amb el missatge:

**THIS IS AN UGLY DEBUG TOOL!**  
To browse the database, follow the link [db](#)

Ugly Browser version: \$Revision: 1.13 \$

**Nota:**

Degut a que la comanda *xindice* que s'utilitza com a eina de treball intenta connectar amb el port 8888 de localhost es pot canviar el port de connexió que s'utilitza en arrencar el servidor Tomcat per aquest nou port, realitzant la modificació d'aquest paràmetre dins dels l'arxius *server-minimal.xml* i *server.xml* ubicats a *\$CATALINA\_HOME/conf*, per exemple mitjançant els editors *vi* o *gedit*. Això afectarà també a les proves realitzades amb eXist.

# Capítol 5 Aplicació Client per a eXist

## 1. Introducció

La classe `InteractiveClient` es pot llençar de dues maneres: primerament es pot utilitzar el script `bin/client.sh`. En segon lloc es pot utilitzar el propi mecanisme de llençament de l'arxiu `start.jar`. Per a seleccionar al client es passa la seqüència `client` com a primer argument, p.e.

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

El client ofereix tres modes d'operació:

1. Si una acció s'especifica a la línia de comandes es processarà de manera no interactiva i el client finalitzarà després de completar-la.
2. Sense cap acció el client canvia a modes interactiu i s'exhibeix una interfície gràfica. En aquesta interfície gràfica ens convida a autenticar-nos. Si entrem per primera vegada com a nom d'usuari "admin" i sense cap paraula de pas
3. Si l'opció és `-l` o `-o` no s'especifica sense cap acció el client canvia a mode shell i convida a l'usuari a entrar sense exhibir cap GUI.

## 2. Configuració general

El client llegeix les opcions implícites de l'arxiu `client.properties` ubicat a `exist/WEB-INF`. La majoria de les característiques es poden sobreescrivre per paràmetres de la línia de comandes o fent `set command` en mode shell.

El client confia en l'API XML:DB per a comunicar amb la base de dades. Així treballarà tant en remot com per instàncies de la base de dades encaixades en programari. La instanciació correcta de la base de dades està determinada per la base URI XML:DB com s'especifica en les propietats de l'arxiu o a través de les opcions de la línia de comandes.

La URI XML:DB base usada pel client per les connexions està definida per la `uri=` propietat. Per defecte, aquesta és assignada a `uri=xmldb:exist://localhost:8080/exist/xmlrpc`. El client intentarà connectar amb una instància de la base de dades que funciona dins del servidor web en el port local 8080 del host. Això no significa que el client no s'està comunicant a través de la xarxa. De fet, qualsevol URI BD:XML conté una part host que és accessible pel protocol XMLRPC.

Hi ha diferents maneres d'especificar una URI XML:DB base alternativa:

Primerament cal canviar la `uri=` propietat de `client.properties`.

Segon, es pot usar el paràmetre `-o` des de la línia de comandes per a seleccionar temporalment un altre destí per a la connexió. Per exemple, començar el client en modus local fer:

```
./client.sh -ouri=xmldb:exist://
```

Per accedir a un servidor funcionant dins del webserver Jetty al port 8088 d'un host remot cal fer

```
./client.sh -ouri=xmldb:exist://localhost:8088/exist/xmlrpc
```

En canvi per accedir al servidor eXist funcionant dins del webserver Tomcat al port 8888 d'un host remot cal fer. Aquesta instrucció serà la que tindrà efecte per accedir a la base de dades de l'aplicació final.

```
./client.sh -ouri=xmldb:exist://www.elmeuservidor.com:8888/exist/xmlrpc
```

Si s'ha especificat una paraula de pas per l'usuari `admin` (de la manera com es descriu en el document de seguretat), ens podem autenticar-nos a la base de dades. En aquest cas cal especificar `-u username` a la línia de comandes, p.e.

```
./client.sh -l -u joan -P meupass ouri=xmldb:exist://www.elmeuservidor.com:8888/exist/xmlrpc
```

si l'opció `-P meupass` manca el client ens la demanarà.

### 3. Fent ús de la shell

No pretenc descriure la shell detingudament aquí. En prémer ? dins la consola de la shell apareix el llistat de comandes:

```
type help or ? for help.
```

```
exist:/db> ?
```

```
--- general commands ---
```

```
ls (llista col·leccions contingudes)
```

```
cd [collection].] ( canvia la col·lecció activa)
```

```
.....// tot el llistat de comandes//.....
```

```
unlock resource desbloqueja per a l'escriptura el recurs especificat .
```

```
quit ( surt del programa)
```

El mode shell permet consultar l'història de les comandes executades depenent del tipus de sistema operatiu. Als sistemes Unix intentarà carregar la biblioteca del readline del GNU, que es part de la majoria d'instal·lacions Unix. Això dona accés a tot el que podem fer amb la shell de Linux. Per exemple, al prémer la tecla tabulador intentarà finalitzar el nom de la col·lecció i del document. Per a que això funcioni, la biblioteca *nativa lib/core/libJavaReadline.so* deu ser trobada pel carregador del sistema. A Linux tant sols cal afegir *lib/core* a la variable `LD_LIBRARY_PATH` (l'script *client.sh* fa això automàticament ).

Per explicar el mode shell segueixo el següent exemple per a veure com s'emmagatzement els arxius a la base de dades.

Picant *mkcol centre* i prement enter crearem una "col·lecció cursos"- *collection* dins la qual podem posar alguns dels documents exemple provistos amb eXist o d'altres documents o subcol·leccions que correspondrien als de l'aplicació pràctica final.

Per a comprovar si la nova col·lecció és present cal fer *ls* per a veure una llista del continguts de les col·leccions actuals la llista següent mostra un exemple de com afegir els documents exemple per a la pràctica :

#### **Exemple: Afegint els documents exemple**

```
exist:/db>mkcol centre
created collection.
exist:/db>cd centre
exist:/db/centre>mkcol alumnes
created collection.
exist:/db/alumnes>cd alumnes
exist:/db/cursos/alumnes>put exemples/
storing document joan.rdf (1 of N) ...done.
.....
storing document angels.rdf (N of N) ...done.
exist:/db/centre/alumnes> cd /
```

Afegir arxius a la base de dades es fa fent servir *put*. Put accepta un únic arxiu, un patró d'arxius o el nom d'un directori com argument. Si és un directori tots els arxius XML i XSL del directori seran afegits a la base de dades.

Per afegir els arxius en el directori exemples/ solament cal introduir

```
put centre/.
```

Per a veure si els arxius estan emmagatzemats es pot veure el seu contingut mitjançant la comanda *get* p.e. :

```
get joan.rdf
```

**put** també accepta patrons d'arxius com p.e. amb comodins ? o \*, \*\*.

```
put /root/centre/**/* .rdf
```

L'ordre anterior analitzarà qualsevol arxiu RDF trobat en el directori exemples i dins de qualsevol dels seus subdirectoris.

### Nota

Si quan executem mentre experimentem amb eXist trobem problemes vol dir que els arxius de la base de dades són corruptes llavors caldrà eliminar els arxius creats i tornar a treballar . Els arxius de dades tenen extensió .dbx. Es poden trobar dins del directori webapp/WEB-INF/data or WEB-INF/data, dependent de la nostra instal·lació . Això també és vàlid per a fer backup d'aquests arxius i posteriorment restore en cas de que la base de dades estigui corrupte.

## 4. Especificant Paràmetres

---

El client utilitza la biblioteca CLI del projecte Excalibur d'Apache per analitzar els paràmetres passats a la línia de comandes. Això significa que s'utilitzen les mateixes convencions per a la majoria de les eines del GNU. La majoria dels paràmetres tenen una forma curta i un altre de llarga: per exemple, l'usuari pot ser especificat amb la forma curta **-u user** o en la forma llarga **-user=user**. També podem combinar paràmetres curts: per exemple, **-ls** és la forma curta de **-l -s**.

## 5. Emmagatzemar documents

---

Per emmagatzemar un joc de documents cal fer servir els paràmetres **-m** i **-p** , p.e.

```
bin/client.sh -m /db/centre/alumnes -p /home/exemples/alumnes
```

L'argument **-m** es diferencia de l'opció **-c**, per que li diu implícitament que crei una col·lecció perduda. El paràmetre **-p** significa que tots els arguments següents es deuen interpretar com una llista de paths a documents XML, és a dir, permet especificar més d'un document o directori.

Si el path passat denota a un directori, el client intentarà emmagatzemar tots els documents que hi hagin dins aquest directori a la base de dades. En canvi això no és recursiu pels subdirectoris. Per això caldrà passar-li el paràmetre **-d**. Per exemple,

```
bin/client.sh -d -m /db/centre -p /home/exemples
```

farà recursiva l'acció per a tots els subdirectoris que pengin de /home/exemples. Per a cada subdirectori, serà creada una col·lecció de l'arrel de la col·lecció de /db/centre, d'aquesta manera es pot usar aquesta opció per importar recursivament un arbre de col·leccions sencer.

## 6. Esborrant Col·leccions/Documents

---

Els paràmetres **-r** and **-R** es poden usar per a esborrar un document o una col·lecció. **-r** elimina un sol document de la col·lecció especificada pel paràmetre **-c**. Per exemple,

```
bin/client.sh -c /db/centre/alumnes -r joan.rdf
```

elimina el document joan.rdf de la col·lecció /db/centre/alumnes. Per eliminar la col·lecció sencera cal fer

```
bin/client.sh -c /db/centre -R alumnes
```

## 7. Executant consultes

---

Per executar consultes cal utilitzar el paràmetre **-x**. Aquest paràmetre accepta un argument opcional el qual especifica la consulta a executar. En canvi si es passa la consulta XQuery des de la línia de comandes pot presentar un problema en alguns sistemes operatius per que la shell pot interpretar els espais en blanc com a separadors de caràcters ( com passa a Windows). D'aquesta manera si no es passa cap argument a **-x** el client intentarà llegir la pregunta per l'entrada estàndard. Per exemple a Unix es pot fer

```
echo "//SPEECH[LINE &= 'love']" | bin/client.sh -x
```



les consultes també es poden llegir des de un arxiu. Per exemple,

```
./client.sh -F ../samples/xquery/fibo.xq
```

executa la XQuery continguda a fibo.xq.

Hi ha un paràmetre addicional que s'executa junt `-x`, aquest és `-n` que especifica el nombre de respostes que es deuen imprimir per la sortida estàndard.

Aquestes opcions són molt farragoses i es millor utilitzar la GUI que incorpora el client.

## 8. Fent servir una configuració de la BD:XML alternativa

Si inicialitzem el client amb l'opció `-l` o `-ouri=xmldb:exist://`, això llençarà el propi cas encaixat de la base de dades. Per defecte aquesta configuració es llegeix de l'arxiu `conf.xml` ubicat a `EXIST_HOME`.

En canvi si es vol utilitzar un altre instal·lació de la base de dades, emmagatzemada en una localització alterna es pot fer amb l'opció `-C` i a continuació el paràmetre de configuració corresponent:

```
bin/client.sh -C /home/exist/test/conf.xml
```

Això llençarà temporalment un nou cas de la bbdd. Obviament la opció `-C` implica la opció `-l`.

## 9. Backup/Restore

### Backup/Restore amb la línia de comandes

La revisió 0.9.1 conté un client senzill de backup/restore que es llença mitjançant el script **bin/backup.sh**. El client permet fer backup de qualsevol col·lecció local o remota disponible amb l'API XML:DB API. Per fer un backup d'una col·lecció cal fer

```
bin/backup.sh -d backup -u admin -p somepass -b /db -ouri=xmldb:exist://
```

Això llençarà una instància en mode local de la col·lecció arrel `/db`. Es crearà una jerarquia de directoris dins del directori de backup. Els directoris correspondran a la jerarquia de col·leccions trobada a la base de dades.

L'eina podrà també fer backup de permisos d'usuaris per a cada col·lecció i document.

Aquesta informació s'escriu en l'arxiu especial `__contents__.xml` que es troba dins de cada subdirectori. Es necessita d'aquest arxiu per a restablir el contingut de la base de dades.

Per a restablir dades a las quals s'ha fet un backup cal usar l'opció `-r` i passar un dels arxius `__contents__.xml` com argument. L'eina restablirà tots els arxius i col·leccions descrits en aquest arxiu XML.

No obstant, en la versió d'eXist 1-0b1 amb la que estic probant, aquest script no apareix i per tant les tasques de Backup/Restore s'han realitzat usant la interfície gràfica.

### Backup/Restore fent ús de la GUI

Mitjançant la icona corresponent es realitza mateixa tasca descrita anteriorment. Ens permet escollir la subcol·lecció o la pròpia col·lecció db. Si escollim la col·lecció `/db/centre` i posteriorment com a destinació a `/root/backup` llavors podrem observar que en aquest path s'ha enregistrat un arbre de directoris des de la col·lecció arrel fins i tot les que es troben dins de la col·lecció esmentada, però en canvi les seves col·leccions "germanes" no. Això permet doncs fer còpies de seguretat parcials. A més, a partir del directori EDI en tots trobem un arxiu `__contents__.xml` que mostra p.e. el següent:

```
< collection name="/db/centre" owner="admin" group="dba" mode="755">
  <subcollection name="cursos" />
  <subcollection name="alumnes" />
</collection>
```

Quan volem restablir la base de dades llavors escollim la icona corresponent l'aplicació client ens permetrà escollir un arxiu. `__contents__.xml` del qual llegirà la informació mostrada anteriorment i que seguirà de patró i de manera recursiva per a crear dins de la base de dades, totes les subcol·leccions i documents XML especificat en ell. Si actualitzem la pantalla podrem veure que s'han restablert les col·leccions.

# Capítol 6 Aplicació client per a Xindice

## 6.1. Administració de la Bases de Dades

L'administració de la Base de dades de Xindice es realitza des de la línia de comandes utilitzant la comanda *xindice*. Aquesta comanda permet interactuar i alterar la configuració de la base de dades mentre el sistema està funcionant.

### 6.1.1 Gestionar Col·leccions

#### Llistar col·leccions

Llista totes les col·leccions sota el context determinat.

##### Sintaxi:

```
xindice list_collections -c (or context) [-v (or )]
```

Acció	<i>list_collections</i>
Acció en forma abreujada	<i>lc</i>
Nivell d'accés	<i>User</i>

##### Paràmetres:

- -c El context col·lecció sota el qual seran llistades totes les col·leccions
- -v Mostra sortida

##### Exemples:

Podem llistar totes les col·leccions filles que s'ubiquen sota la col·lecció */db*.

```
./xindice.sh lc -c /db
```

#### Afegir una col·lecció

Afegeix una nova col·lecció a la base de dades. Quan afegim una col·lecció sota una jerarquia de col·leccions preexistent totes les col·leccions pares continuaran existint a la base de dades.

##### Sintaxi:

```
xindice add_collection -c (or context) -n (or name) [-v (or )]
```

Acció	<i>add_collection</i>
Acció en forma abreujada	<i>ac</i>
Nivell d'accés	<i>Admin</i>

##### Paràmetres:

- -c El context col·lecció sota el qual crearem la nova col·lecció.
- -n El nom de la col·lecció a crear
- -v Mostra sortida

##### Exemples:

Afegim una col·lecció anomenada *cursos* dins a la col·lecció */db*. Posteriorment anem afegint altres col·leccions

```
./xindice.sh ac -c /db -n centre
```

```
./xindice.sh ac -c /db/centre -n cursos
```

```
./xindice.sh ac -c /db/centre -n alumnes
```

#### Esborrar una col·lecció

Esborra una col·lecció o subcol·lecció de la base de dades. Si esborrem una col·lecció que té subcol·leccions i/o documents, tots ells seran esborrats.

### Sintaxi:

```
xindice delete_collection -c (or context) -n (or name) [-v (or )]
```

Acció	<i>delete_collection</i>
Acció en forma abreujada	<i>dc</i>
Nivell d'accés	<i>Admin</i>

### Paràmetres:

- -c El context de la col·lecció sota el qual esborrarem la col·lecció esmentada.
- -n El nom de la col·lecció a borrar
- -v Mostra sortida

### Exemples:

Eliminem una col·lecció  `cursos` ubicada dins de a col·lecció `/db/centre`.

```
./xindice.sh dc -c /db/centre -n cursos
```

## 6.1.2 Gestionar Indexos

El sistema d'indexació de Xindice defineix índexos per una resposta més ràpida a les consultes XPath que s'usen de manera comuna. Si no es defineixen, llavors el rendiment es veurà prou afectat ja que tindrà que scannejar la col·lecció sencera per a crear el resultat del joc de nodes.

La indexació es pot realitzar fent servir la comanda `xindice`

### Afegir un Índex

Fent servir un arxiu XML com el que mostrem a continuació podrem afegir un índex a l'element `product_id` ja que les cerques pels productes utilitzant el `product_id` són comunes

```
<?xml version="1.0"?>
<product>
  <product_id>120302</product_id>
  <description>Glazed Ham </description>
</product>
```

Això ho podem fer mitjançant l'esmentada comanda de manera que si nosaltres volem crear un índex anomenat `idindex` per a tots els elements `product_id` de la col·lecció `/db/data/catalog`.

```
xindice add_indexer -c /db/data/catalog -n idindex -p product_id
```

Una vegada fet això, el motor de consultes podrà usar aquest índex per a resoldre consultes XPath que impliquin una restricció pel valor `product_id` de l'element

El paràmetre `-p` de la comanda especifica el patró que usará en la indexació. Aquests patrons són usats pel sistema d'indexació per a determinar el millor compliment, una indexació basat en parelles i l'actualització dels índexos. El patró usat cal que s'assembli al següent esquema

Patró	Descripció
<code>elem</code>	Valor de l'element anomenat
<code>elem@attr</code>	Valor de l'atribut de l'element anomenat
<code>*</code>	El valor de tots els elements
<code>*@attr</code>	Valor de l'atribut anomenat de tots els elements
<code>elem@*</code>	Valor de tots els atributs de l'element anomenat
<code>*@*</code>	Valor de tots els atributs de tots els elements

Si es vol indexar un espai de noms diferent del que hi ha per defecte caldrà incloure l'emplaçament URI entre claudàtors.

```
[http://www.world.org/People]person
*@[http://www.world.org/People]id
[http://www.world.org/People]person@[http://www.world.org/People]id
```

No podem incloure un prefix a aquests patrons, com un sistema d'indexació, Degut a que la majoria de les aplicacions que processen espais de noms, processen els elements de l'espai de noms i els atributs de manera independent del prefix que s'ha usat.

### Indexar tant atributs com elements (ambdós)

Degut a que els patrons poden reconèixer tant elements com atributs, però no ambdós, per ordenar indexar tots els elements i valors d'atributs d'una col·lecció es poden crear dues entrades per l'índex. El patró \* indexarà tots els elements i el patró\*@\* indexarà tots els atributs de tots els elements .

```
xindice add_indexer -c db/data/catalog -n idindex -p '*'
xindice add_indexer -c db/data/catalog -n idindex -p '*@*'
```

Un excessiu ús d'indexos amb comodí pot afectar al rendiment del sistema d'indexació. El millor consell serà doncs especificar indexos d'elements o atributs sempre que sigui possible i solament definir indexos amb comodí quan sigui absolutament necessari.

## 6.2 Manipulació de documents

El servidor Xindice proveeix de mecanismes per emmagatzemar, recuperar i eliminar documents XML. En tots els exemples assumeixo que executem Xindice des de el directori WEB-INF .

### 6.2.1 Afegir Documents

Afegeix un document a una col·lecció o subcol·lecció. Afegir un document requereix dos paràmetres – la col·lecció sota la que s'emmagatzemarà i la via d'accés del document. Si no es proveeix al document d'una clau llavors s'usarà una clau autogenerada pel sistema. Els documents que no siguin afegits a col·leccions es considerarà que no existiesen. Cuan introduïm la ruta d'accés de l'arxiu hem d'assegurarnos d'incloure el path i l'extensió de l'arxiu.

#### Sintaxi:

```
xindice add_document -c (or context) -f (or file path) [-n (or key to assign to document)] [-v (or )]
```

Acció	<i>add_document</i>
Acció en forma abreujada	<i>ad</i>
Nivell d'accés	<i>User</i>

#### Paràmetres:

- -c El context col·lecció sota el qui afegirem el doument.
- -f La via d'accés completa de l'arxiu del document que serà afegit
- -k La clau que assignem al document.
- -v Mostra sortida

#### Exemples:

##### Afegir documents amb una clau

El document joan.rdfxml serà afegit a la col·lecció /db/centre/alumnes i serà emmagatzemat sota la clau joan.

```
xindice add_document -c /db/centre/alumnes -f joan.rdf -n joan
```

##### Afegir documents sense una clau

El document joan.rdf serà afegit a la /db/alumnes. Si no es proveeix de cap clau, aquesta es generarà pel servidor i tindrà un aspecte similar a això 0625df6b0001a5d4000bc49d0060b6f5

```
xindice add_document -c /db/centre/alumnes -f joan.rdf
```

## 6.2.2 Afegir múltiples documents

Afegeix múltiples documents a una col·lecció o subcol·lecció. Aquesta comanda requereix dos paràmetres – la col·lecció sota la que s'emmagatzemaran i la via d'accés dels documents. Els documents poden ser afegits a les col·leccions i també les subcol·leccions tantes com existeixin. Als documents s'els hi assignarà com a clau de document el seu nom d'arxiu. El paràmetre opcional d'extensió podrà ser usat per importar documents amb una certa extensió. Les claus dels documents seran mostrades quan siguin creats

### Sintaxi:

```
xindice add_multiple_documents -c (or context) -f (or Directory to use) [-e (or file extension)] [-v (or )]
```

Acció	<i>add_multiple_documents</i>
Acció en forma abreujada	<i>addmultiple</i>
Nivell d'accés	<i>User</i>

### Paràmetres:

- -c El context col·lecció sota el qui afegirem els documents.
- -f La via d'accés completa del directori d'on importem els documents
- -e L'extensió dels arxius que farem servir per importar els documents.
- -v Mostra sortida

### Exemples:

Afegim tots els arxius desde un directori a la col·lecció

```
xindice addmultiple -c /db/cursos -f /tmp/mydocs
```

Afegint tots els arxius desde un directori amb l'extensió de ".xml"

```
xindice addmultiple -c /db/cursos -f /tmp/mydocs -e xml
```

## 6.2.3 Recuperar documents

Els documents poden ser recuperats des de la base de dades fent servir el ID que es va assignar.

### Recuperant un document fent servir un ID

El document identificat per la clau 000001 serà recuperat des de la col·lecció /db/cursos/alumnes i emmagatzemat en l'arxiu joan.rdf

```
xindice retrieve_document -c /db/cursos/alumnes -f 000001 -n joan.rdf
```

## 6.2.4 Eliminant documents

### Eliminant un document fent servir un ID

El document identificat per la clau 000001 serà eliminat de la col·lecció /db/cursos/alumnes.

```
xindice delete_document -c /db/cursos/alumnes -n 000001
```

## 6.3. Administració del Servidor

### Engegar el servidor

Si assumim que tenim instal·lat Xindice sota Tomcat i que s'ha definit la variable d'entorn CATALINA\_HOME en el directori d'instal·lació de Tomcat

```
cd $CATALINA_HOME/bin
```

```
./startup.sh
```

## Aturar el servidor

Per aturar el servidor Xindice cal fer-ho de la següent manera , assumim també que estem fent servir Tomcat

```
cd $CATALINA_HOME/bin
./shutdown.sh
```

## Backup de les nostres Dades

### Backup del servidor

Tant sols cal aturar el servidor de l'aplicació i copiar l'estructura del directori de la base de dades, p.e. fent servir Tomcat i la la versió server de xindice amb la configuració per defecte :

```
catalina.sh stop
cd $CATALINA_HOME/webapps/xindice/WEB-INF
cp -pr db /backup/db
catalina.sh start
```

### Restablir les dades

Restablir les dades és sencillament eliminar la base de dades actual i revertir el procés de backup. En aquest cas fent servir Tomcat això seria d'aquesta manera :

```
catalina.sh stop
cd $CATALINA_HOME/webapps/xindice/WEB-INF
rm -rf db
cp -pr /backup/db db
catalina.sh start
```

## Exportant el contingut de la Base de Dades

Xindice inclou eines per exportar les dades a un directori jeràrquic i també importar les dades desde un directori jeràrquic. Cada directori jeràrquic correspon a una col·lecció a Xindice. Cada document XML és emmagatzemat com un nom separat com la clau desde la base de dades.

### Exportar la base de dades

En aquest exemple s'assumeix que el directori *xindice/bin* es troba en el teu path.

```
xindice export -c /db/cursos -f /path/to/data
```

El contingut sencer de la col·lecció */db/cursos* serà exportat al directori */path/to/data*.

### Importar la base de dades

En aquest exemple assumim que el directori *xindice/bin* és en el teu path.

```
xindice import -c /db -f /path/to/data/cursos
```

cada directori sota */path/to/data/cursos* serà usat per a crear una col·lecció i tots els documents XML dins de la jerarquia seran importats a la base de dades. Es pot també restringir els documents que seran importats afegint *-i* i la extensió dels arxius que volem importar.

## 6.4 Consultar la base de dades

Xindice actualment suporta XPath com a llenguatge de consulta. A Xindice les consultes poden ser executades a nivell de document o a nivell de col·lecció de manera que la consulta s'executarà en molts documents i el joc de resultats contindrà tots els nodes vàlids de tots els documents de la col·lecció. El servidor Xindice server també suporta la creació d'índexos en documents XML que acceleraran les consultes XPath. Cal donar un cop d'ull a [Administrators Guide](#) per a veure més detalls de com configurar índexos.

Podem executar una consulta XPath fent servir la línia de comandes i el resultat de la consulta serà visualitzat això ho veurem millor en el següent capítol.

Per a més informació caldrà veure la [guia de desenvolupadors](#).

# Capítol 7 XPath

## 7.1 Característiques generals d'XPath

El W3C defineix [XPath](#) com un mitjà per a seleccionar qualsevol unitat simple de contingut, és a dir qualsevol node dins d'un document XML.

Els nodes es seleccionen utilitzant un constructor anomenat expressió d'XPath, que sovint es pot descomposar en expressions subordinades. Una expressió completa d'XPath es codifica com un *trajecte de cerca*, que es compon d'un o de varis *passos de cerca* delimitats per *barres inclinades* cap a la dreta (/). Un trajecte de cerca senzill s'assembla a la ruta d'accés d'un arxiu determinat per un sistema d'arxius o un servidor. La barra inclinada inicial comença la cerca en l'arrel del document. Cada pas de cerca amb èxit llegit d'esquerra a dreta ens endinsa en el contingut de l'arbre del document, així cadascun d'ells es considera en el contexte dels precedents. Cal dir que el contexte d'un node li dona al node un significat diferent. En el cas de l'aplicació desenvolupada (veure capítol 10), per exemple, *Person* pot definir, en un document de la col·lecció alumnes,,al propi alumne o a una perosna coneguda per aquest si està dins de l'element foaf:knows.

El node que estableix el contexte en que s'avalua una expressió XPath s'anomena *node contexte*. La sintaxi d'un *pas de cerca* consta de tres components: *eix*, *comprovació de node* i *predicat* amb el següent format:

*eix::comprovació\_de\_node[predicat]*

L'objecte d'aquests tres components consisteix en seleccionar un node, o un conjunt de nodes (*node-set*) que tingui certa relació amb el node contexte.

- [L'eix](#) : Li indica al processador en quina direcció cal mirar a partir del node contexte. Existeix una sintaxi abreujada pels eixos.
- [Comprovació de node](#): La part de comprovació de node d'una expressió XPath o d'un pas de cerca limita la vista de nodes disponibles al llarg d'un eix determinat a solament un subconjunt de tots ells, és a dir genera un conjunt de nodes candidats en aquesta expressió de pas.
- [El predicat](#): Aquesta part és opcional i refina encara més el conjunt de nodes candidats que estableix el eix i la comprovació de node ja que el que apareix entre claudàtors del predicat és una *comprovació lògica* que retorna un valor veritat o fals i per tant solament s'escullen els candidats en el que el resultat és veritat.

A més de la seva posició en l'arbre del document, tot resultat que retorna una expressió d'XPath té un valor. Aquest, anomenat valor de node pot ser de diferents tipus de dades:

- Node-set: el resultat és una llista desordenada d'un o més nodes.
- String: el resultat és una cadena de text
- Numèric: el resultat és un nombre en coma surant.
- Booleà. El resultat és un valor veritat o fals.

El conjunt de funcions XPath definit permet la manipulació del contingut de maneres diverses retornant per tant, subconjunts dels nodes seleccionats pels trajectes de cerca. Aquestes es poden classificar, segons el tipus de dada del valor de node que retorna, en:

- [Funcions de conjunts de nodes](#)
- [Funcions de cadena](#)
- [Funcions booleanes](#)
- [Funcions numèriques](#)

Les consultes XPath i els resultats descrits en l'ús d'XPath a Xindice i a eXist funcionaran sobre un exemple que correspon a l'aplicació pràctica que s'està desenvolupant partint del conjunt de col·leccions i de l'afegit de documents en elles segons el capítol anterior i dissenyats en el capítol 10.



## 7.2 Ús d'XPath a Xindice

Xindice actualment suporta XPath com a llenguatge de consulta. Les consultes poden ser executades des d'una aplicació client (caldrà donar un cop d'ull a la [developers guide](#)), mitjançant l'interfície XML-RPC, o per la línia de comandes (donar un cop d'ull a la [command line tool guide](#)). A continuació s'exposa una petita descripció de quines consultes XPath suporta i quins són els resultats d'aquestes consultes, per a fer una idea del que es pot fer des de la línia de comandes

### 7.2.1 Consulta de documents

Un dels usos més comuns de XPath és el fet d'obtenir documents que satisfacin algun criteri. A continuació exposo algun exemple per a veure el seu funcionament

La següent instrucció retorna una cadena de text de tots els elements continguts dins els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* és "XML" o "BBDD"

```
xindice xpath -c /db/centre/alumnes -q "string(//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"])"
```

Aquesta mostra el node text de l'element *foaf:name* els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* "XML" o "BBDD"

```
xindice xpath -c /db/centre -q "//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"]/foaf:name/text()"
```

També estan suportades les expressions XPath amb resultat numèric. Per donar un exemple la següent instrucció compta els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* és "XML" o "BBDD"

```
xindice xpath -c /db/centre/* -q "count(//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"])"
```

## 7.3 Ús d'XPath a eXist

Degut a la comoditat de treball que ofereix la interfície gràfica d'usuari d'eXist he desenvolupat més exemples d'XPath per a estudiar el funcionament de les funcions.

- Selecciona l'element */rdf:RDF* que es troba en el primer nivell de descendència

```
/rdf:RDF
```

- Selecciona tots els elements *foaf:Person*

```
//foaf:Person
```

- Selecciona tots els elements descendents de cada element *foaf:Person*

```
//foaf:Person/*
```

- Selecciona de cada document i de tots els elements *foaf:Person* el primer que es troba en cada document, és a dir el que defineix a la persona que describem segons el model d'arxiu fins l'últim descendent.

```
//foaf:Person[1]
```

- Selecciona de cada document i de tots els elements *foaf:Person* el primer que es troba en cada document, és a dir el que correspon a la segona persona coneguda de la persona que describem segons el model d'arxiu.

```
//foaf:Person[3]
```

- Selecciona de cada document i a l'últim element *foaf:Person* dins de l'arbre i que correspon a la descripció del gerent de la última empresa enregistrada en la que ha treballat.

```
//foaf:Person[last()]
```

- Selecciona tots els atributs *rdf:nodeID* dels elements *foaf:Person*

```
//foaf:Person/@rdf:nodeID
```

- Mostra l'element *foaf:givenname* els elements *foaf:Person* que tenen atribut *rdf:nodeID*

```
//foaf:Person[@rdf:nodeID]/foaf:givenname
```

- Selecciona l'element *foaf:Person* que tingui pel seu atribut *rdf:nodeID* el valor *000003*  
`//foaf:Person[@rdf:nodeID="000003"]`
- Selecciona tots els elements *foaf:Person/foaf:givenname* en els que el valor de l'atribut *rdf:nodeID* de l'element *foaf:Person* sigui *000003*  
`//foaf:Person[@rdf:nodeID="000003"]/foaf:givenname`
- Selecciona tots els elements *foaf:Person* en els que el valor de l'atribut *rdf:nodeID* sigui *000003* i mostra el node text de l'element *foaf:givenname*  
`//foaf:Person[@rdf:nodeID="000003"]/foaf:givenname/text()`
- Mostra el node text de l'element *foaf:name* els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* "XML" o "BBDD"  
`//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"] /foaf :givenname /text()`
- Compta els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* és "XML" o "BBDD"  
`count(//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"])`
- Retorna una cadena de text de tots els elements continguts dins els elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* és "XML" o "BBDD"  
`string(//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"])`
- Retorna un booleà si existeix algun elements *foaf:Person* en els que el predicat de l'element *foaf:topic\_interest* és "XML" o "BBDD"  
`boolean(//foaf:Person[foaf:topic_interest="XML" or foaf:topic_interest="BBDD"])`
- Retorna els elements *foaf:knows* que tenen menys de dos elements fills de tipus *foaf:Person*. És a dir persones que tenen menys de dos amics  
`//foaf:Person/foaf:knows[count(foaf:Person)<2]`
- Retorna el predicat corresponent a *foaf:name* de les persones en les que els elements *foaf:knows* tenen menys de dos elements fills de tipus *foaf:Person*. És a dir nom de les persones que tenen menys de dos amics  
`//foaf:Person/foaf:knows[count(foaf:Person)<2]/parent::* /foaf:givenname/text()`
- Troba tots els elements que el seu nom d'element comenci pel prefixe "rdf"  
`//*[starts-with(name(),"rdf")]`
- Troba tots els elements que el seu nom d'element tingui la cadena "knows"  
`//*[contains(name(),"knows")]`

# Capítol 8 XQuery

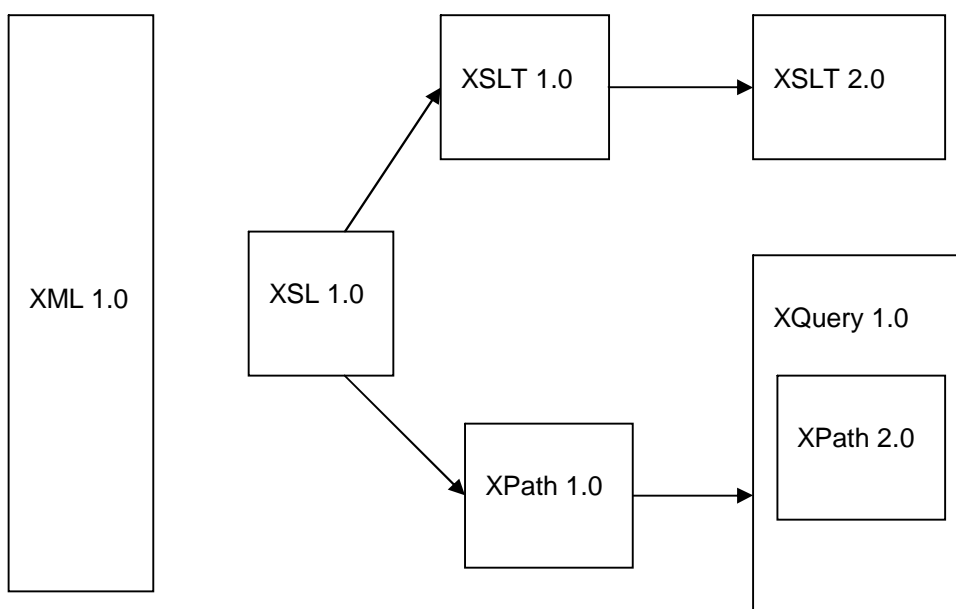
## 8.1. Característiques generals de XQuery

### Introducció.

XQuery és un llenguatge de consulta estàndard publicat pel W3C i dissenyat per a processar dades XML.

La capacitat i flexibilitat de XQuery i en l'ús intensiu que fa de XPath el converteixen en el llenguatge de consulta per a usar-se sobre entorns i documents XML així com un excel·lent llenguatge per realitzar consultes en sistemes de BD:XML..

En la següent figura podem observar algunes de les especificacions del W3C. Com podem observar l'especificació 1.0 de XPath va pertànyer a XSL 1.0 i posteriorment es va desenvolupar com una especificació separada. Actualment XPath 2.0 s'ha desenvolupat paral·lelament junt a l'especificació 1.0 de XQuery.



Query Working Group, creat en octubre de 1999 pel W3C, és el grup encarregat de la elaboració del llenguatge de consulta XQuery per a XML

XQuery està encara en procés d'elaboració

S'han publicat diferents esborranys de treball:

[XQuery 1.0: An XML Query Language,](#)

[XML Query Requirements](#)

[XQuery 1.0 and XPath 2.0 Data Model](#)

[XQuery 1.0 Formal Semantics](#)

[XQuery 1.0 and XPath 2.0 Functions and Operators](#)

[XML Query Use Cases](#)

### El futur de XQuery.

Sembla que en un futur XQuery serà el llenguatge de consulta universal per a treballar amb l'entorn de XML ja que obre noves possibilitats per a consultar magatzems de dades :

- XML i fragments de XML en tota classe de documents emmagatzemats en BD:XML.
- XML enllaçat en sistemes d'administració de bases de dades relacionals.

## Expressions FLWOR

Les expressions **FLWOR** (*anomenades expressions FLOWER*) són al XQuery el que les distintes clàusules dins d'una sentència **Select** (*select, from, where, etc*) són al SQL. És a dir, són els seus blocs principals. El nom li ve de **For, Let, Where, Order by** i **Return**:

Una expressió FLWOR

- Comença amb una o més clàusules for o let en cualsevol ordre (com a mínim cal que n'hi hagi un )
  - • Seguides per una clàusula where (opcional)
  - • Pot aparèixer una clàusula sort by (opcional)
  - • Finalitza amb una clàusula return (obligòtria)
- FOR: associa una o més variables a expressions, creant un conjunt de tuples en el que cada tupla vincula una variable donada a un dels ítems a los qui està associada l'expressió avaluada.
  - LET: vincula les variables al resultat d'una expressió, afegint aquests vincles a les tuples generades per la clàusula FOR.
  - WHERE: filtra tuples, quedant solament aquelles que compleixen amb la condició.
  - ORDER BY: ordena les tuples en el conjunt de tuples.
  - RETURN: construeix el resultat de l'expressió FLWOR per a una tupla donada.

Les clàusules FOR i LET carreguen el conjunt de tuples sobre la qui s'avaluarà la sentència FLWOR, al menys una d'aquestes clàusules te que existir en una consulta. Amb aquestes sentències s'aconsegueix bona part de la diferència entre XQuery de XPath. Entre d'altres coses, permet construir el document que serà la sortida de la sentència.

Algun exemples, basats en la descripció de l'aplicació final i que permeten aclarir quin com s'executen consultes mitjançant XQuery poden ser els següents

- Cerca l'adreça on s'emmagatzema la fotografia de l'alumne identificat com 000001

```
for $a in /rdf:RDF/foaf:Person[@rdf:nodeID='000001']
  let $matric:=$a/@rdf:nodeID
  let $foto:=string($a/foaf:depiction/foaf:Image/@rdf:resource)
  return
    <foto>{$foto}</foto>
```

resultat

```
<foto>http://www.elmeuservidor.com/imatges/joan.jpg</foto>
```

- Cerca la fitxa personal de l'alumne identificat com 000001

```
for $a in /rdf:RDF/foaf:Person[@rdf:nodeID='000001']
  let $matric:=string($a/@rdf:nodeID)
  let $nom:=$a/foaf:givenname/text()
  let $cognom:=$a/foaf:surname/text()
  let $genere:=$a/foaf:gender/text()
  let $self:=string($a/foaf:phone/@rdf:resource)
  let $web:=string($a/foaf:homepage/@rdf:resource)
  return
    <alumne>{ $cognom," ",$nom," ",$matric," ", $genere," ", $self," ", $web }
    <temes>
    {for $temes in $a/foaf:topic_interest/text()
      return
        <tema>{concat( $temes, "; ")}</tema>}</temes>

    </alumne>
```

resultat

```
<alumne>Perez i Braña Joan Anton 000001 home 934640783
http://www.lamevapagina.com<temes>
  <tema>BBDD; </tema>
  <tema>Java; </tema>
  <tema>Linux; </tema>
</temes>
</alumne>
```

- Cerca dels amics d'un alumne que té com a identificador 000001

```
for $a in /rdf:RDF/foaf:Person[@rdf:nodeID='000001']
  let $matric:=string($a/@rdf:nodeID)
  let $nom:=$a/foaf:givenname/text()
  let $cognom:=$a/foaf:surname/text()
  let $genere:=$a/foaf:gender/text()

  return
    <alumne>{ $cognom, " ", $nom, " ", $matric, " ", $genere}<coneguts>

    {for $b in $a/foaf:knows/foaf:Person

      let $nom2:=$b/foaf:givenname/text()
      let $cognoms2:=$b/foaf:surname/text()

      where exists($b)
      return
        <amic> { $nom2, " ", $cognoms2}    </amic>

    }
  </coneguts></alumne>
```

amb el següent resultat

```
<alumne>Perez i Braña Joan Anton 000001 home<coneguts>
  <amic>Guillermo Escribá Martínez</amic>
  <amic>Antonio Martínez Martínez</amic>
  <amic>Carlos Barber Senao</amic>
</coneguts>
</alumne>
```

## 8.2. Estat Actual

eXist implementa l'actual esborrany de XQuery corresponent a novembre de 2003, a excepció del XML Schema. Actualment, eXist compleix la majoria de [casos d'ús oficials de XQuery](#) (segons l'aplicació Bumblebee, eXist supera el 92% dels casos d'ús de XQuery)

Bàsicament, eXist implementa totes les característiques descrites en l'especificació excepte els següents:

- Paraules clau desconegudes : **typeswitch**, **treat as**, **instance of**, **castable as**.
- Característiques relacionades amb l'esquema: **validate**, **import schema**, **declare validation**.
- Degut a que el suport de l'Schema manca, el valor d'un node és sempre tractat com `xs:untypedAtomic`.
- Collations: especificar un altre collation que la que es pren per defecte no és possible.
- Tipus atòmics `xs:gYear`, `xs:gMonth` etc. No estan reconeguts.

Una altra limitació s'aplica als fragments construïts del document, ja que actualment no és possible fer consultes a documents que s'han construït dins XQuery, p.e. una consulta com la següent farà fallida :

```
let $x := <test>Test</test> return $x/test
```

En aquesta etapa de desenvolupament la construcció a memòria de l'arbre DOM no pot ser consultada de la mateixa manera que la base de dades persistent DOM (DOM a memòria no està encara completada però serà una extensió per una versió futura) .Doncs caldrà emmagatzemar el fragment primerament a la base de dades. Es poden utilitzar les funcions proveïdes [xmldb extension functions](#) per fer això .

## 8.3. Extensions

eXist ofereix un nombre de funcions addicionals i dos operadors addicionals. Aquestes extensions es poden agrupar en tres categories principals :

1. Funcions per a especificar el sistema de document d'entrada per a una expressió.
2. Extensions per a cerques textuas.
3. Funcions per a manipular el contingut de la base de dades.
4. Funcions per a ús general.
5. Funcions de transformacions XSL.
6. Funcions relacionades de l'HTTP.

### 8.3.1. Especificant el joc de documents d'entrada

Una base de dades pot contenir un joc il·limitat de col·leccions i de documents. Si s'utilitza el BD:XML API per a preguntar a la BD, només els documents de la col·lecció actual de BD:XML són processats per defecte. En canvi, hi ha quatre funcions addicionals per a canviar aquest comportament : **doc()**, **document()**, **collection()** i **xcollection()**.

Les funcions **collection()** i **doc()** són funcions estàndards de XQuery/XPath. **xcollection()** i **document()** representen extensions de l'especificació d'eXist.

eXist interpreta els arguments de **collection()** i **doc()** com trajectòries absolutes o relatives, conduint a un altre col·lecció o document dins de la base de dades. El paths relatius es resolen depenent a la característica de la base URI del context estàtic de XQuery, que assenyalava generalment a una col·lecció de la base de dades.

Mentre **doc()** es restringeix a un únic document URI com argument, **document** accepta múltiples paths per a ser inclosos com a joc de nodes d'entrada . En segon lloc, cridant **document()** sense cap argument inclou cada node del document de la instanciació actual de la base de dades.

Per exemple:

```
document('joan.rdf', 'angels.rdf')//foaf:Person/foaf:surname/text()
```

La funció **collection()** especifica la col·lecció de la qual els seus documents s'inclouran dins l'avaluació de la consulta. Per defecte, els documents es troben en subcol·leccions de l'especificada col·lecció inclosa. Com exemple, si tenim tres col·leccions, /db/centre, /db/centre/alumnes and /db/centre/cursos, **collection('/db/centre')** inclourà tots els recursos trobats a /db/centre, /db/centre/cursos i /db/centre/alumnes.

Es pot usar la funció **xcollection()** per canviar aquesta conducta p.e.

```
xcollection('/db/centre')//foaf:Person/foaf:surname/text()
```

solament s'inclouran recursos trobats a /db/centre, però no a /centre/cursos ni a /db/centre/alumnes.

Els exemples segons l'aplicació desenvolupada podrien ser els següents:

En aquest cas es preten obtenir el nom del grup que te per a identificador de node DAI12004-05 (veure disseny de l'aplicació)

Emprant xcollection()

```
for $a in xcollection('/db/centre/cursos')/rdf:RDF/foaf:Group[@rdf:nodeID='DAI12004-05']
  let $grup:=$a/foaf:name/text()
  return $grup
```

Emprant collection()

```
for $a in collection('/db/centre')/rdf:RDF/foaf:Group[@rdf:nodeID='DAI12004-05']
  let $grup:=$a/foaf:name/text()
  return $grup
```

Emprant document() o també doc()

```
for $a in document('DAI12004-05.rdf')//foaf:Group/foaf:name/text()
  let $grup:=$a
  return $grup
```

totes elles amb el següent resultat

1 Desenvolupament d'Aplicacions Informàtiques

### 8.3.2. Cerques textuais

La llibreria de funcions estàndard XPath/XQuery conté la majoria de les funcions de manipulació de seqüències que proporcionen la majoria de llenguatges de programació. En canvi aquestes funcions no són suficients per a la cerca d'un joc de paraules clau o d'una frase dins d'una porció més gran del text o amb contingut barrejat (cerques textuais).

Aquest és un punt feble si cal treballar amb documents del tipus *document-centric*, en comparació amb *data-centric*. Per a molts tipus de documents, les funcions estàndard proporcionades no donaran resultats satisfactoris a nivell de rendiment. Per exemple, si hem llegit quelcom de "XML" i "databases" en un capítol del llibre però no estem segurs d'on es localitza aquest fragment. Fent servir XPath estàndard podem intentar fer la següent consulta:

```
contains(//foaf:givenname/text(), 'Joan') and contains(//foaf:givenname/text(), 'Anton')
```

L'execució de la consulta serà excessivament lenta ja que el motor d'XPath ha d'explorar el contingut sencer dels caràcters que formen tots els nodes del cognom i dels seus descendents. Fins i tot no sabem si hem trobat totes les ocurrencies ja que es podrien trobar començant amb majúscules si es troba a l'inici d'una oració.

eXist ofereix operadors addicionals i extensions de funcions per a un accés eficient, basat en índexos, al contingut complet del text dels nodes. Amb eXist es pot reformular l'anterior consulta de la manera següent

```
//foaf:Person[near(//foaf:givenname/text(), 'Joan Anton?', 50)]
```

Aquesta consulta retornarà tots els nodes que contenen ambdues paraules clau amb l'ordre correcte i amb menys de 50 paraules entre elles. A més el comodí a "Anton?" validarà les ocurrencies com "Anton" i "Antoni".

Com en el present aplicació s'ha treballat amb poques dades la funció near no s'ha utilitzat

En el següent exemple cerquem primerament els nombres de matrícula dels alumnes que formen el grup de node DAI12004-05 dins dels documents de la col·lecció cursos i posteriorment cerquem dins de la col·lecció alumnes aquells que a més de pertànyer al grup contenen una "o" al nom i una "e" al cognom retornant aquests valors. En aquest cas es podria haver especificat la col·lecció db/centre/cursos en el primer for i la col·lecció db/centre/alumnes en el segon for, però s'ha aprofitat que tant sols hi ha dues col·leccions amb documents que tenen diferent estructura.

```
for $grup in /rdf:RDF/foaf:Group
```

```
  let $matric:=$grup/foaf:member/foaf:Person/@rdf:nodeID
  where contains($grup/@rdf:nodeID,'DAI12004-05')
  return
```

```
<llista>
```

```
{for $a in /rdf:RDF/foaf:Person
```

```
  let $nom:=$a/foaf:givenname/text()
```

```
  let $cognom:=$a/foaf:surname/text()
```

```
  let $nummatric:=string($a/@rdf:nodeID)
```

```
  where contains($a/foaf:givenname/text(), 'o') and
```

```
contains($a/foaf:surname/text(), 'e') and $matric=$nummatric
```

```
  order by $a/foaf:surname
```

```
  return
```

```
    <alumne>{$nummatric , " " , $cognom , " " , $nom}</alumne>
```

```
}
```

```
</llista>
```

```

resultat
<llista>
  <alumne>000004 Escrivà; Martínez Guillermo</alumne>
  <alumne>000003 Ferres i Villena Salvador</alumne>
  <alumne>000001 Perez i Braña Joan Anton</alumne>
</llista>

```

### 8.3.3. Manipulant el contingut de la Bases de Dades

Les funcions de l'extensió `xmldb` poden ser usades per a crear noves col·leccions de bases de dades, o emmagatzemar el resultat de la consulta prèviament feta dins de la base de dades. Per exemple si suposem que ens han enviat un arxiu gran que contenia un nombre important d'expedients RDF i no volem emmagatzemar aquests expedients en un sol document si no que el que volem fer és emmagatzemar cada expedient en un propi document llavors el que caldrà serà dividir-lo en petites unitats.

Fer servir el full d'estils de XSLT podria ser una alternativa. En canvi, és una tasca molt intensa pel que fa a la memòria. Per que no fer servir XQuery per fer aquesta tasca? L'script següent de XQuery mostra com dividir un arxiu gran de RDF en un nombre de documents més petits:

#### **Example: Fragmentant un Document**

```

xquery version "1.0";

declare namespace xmldb="http://exist-db.org/xquery/xmldb";
declare namespace util="http://exist-db.org/xquery/util";

xmldb:register-database("org.exist.xmldb.DatabaseImpl", true()),
let $root := xmldb:collection("xmldb:exist:///db", "admin", ""),
    $out := xmldb:create-collection($root, "output")
for $rec in /rdf:RDF/* return
  xmldb:store($out, concat(util:md5($rec/@rdf:about), ".xml"),
    <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
      {$rec}
    </rdf:RDF>
  )

```

Les funcions de l'extensió es basen en DB:XML API, de manera que primerament el que cal fer és enregistrar el driver de la base de dades amb **xmldb:register-database**. Llavors recuperem la col·lecció de l'arrel amb **xmldb:collection**. La variable retornada per **xmldb:collection** es presa com un objecte Java. Nosaltres podem passar aquesta variable a **xmldb:create-collection** per a crear una nova subcol·lecció anomenada "output".

La expressió **for** itera a través de tots els elements fills de l'element de marca `rdf:RDF`. En cada iteració s'usa **xmldb:store** per escriure a en el nou document el node actual fill.

Òbviament es requereix un únic nom per a cada document. En aquest exemple, el URI contingut en l'atribut `rdf:about` és únic, així que calculem una clau MD5 d'ell i afegim ".xml" de manera que el fem servir com a nom del document.

### 8.3.4. Funcions d'ús general

El namespace <http://exist-db.org/xquery/util> conté un nombre de funcions d'utilitat comuns. Per exemple, la funció **util:md5** usada anteriorment està definida aquí.

No explicarem totes les funcions aquí ja que elles estan documentades a ta taula [functions table](#). En canvi hi ha una funció que necessita una certa explicació:

**util:eval** s'utilitza per a executar dinàmicament una expressió construïda dins XQuery dins un script XQuery que s'estigui processant. Això és molt pràctica ja que en una aplicació basada en web, on les consultes es construeixen dinàmicament en funció dels paràmetres que l'usuari ha passat per petició a l'aplicació.



En aquest cas hi hauria una un script XQuery que seria responsable d'avaluar els paràmetres proveïts per l'usuari. Aquest script usa els paràmetres de la petició HTTP per a crear un altre expressió XQuery. Després passa la seqüència construïda de la consulta a **util:eval** per a avaluar l'expressió, després del procés retorna els resultats i els mostra a l'usuari. Més informació respecte a com fer aplicacions web amb XQuery la podem trobar a la [Developer's Guide](#). Aquí a sota hi ha un petit exemple de com sumar o restar dos nombres:

**Exemple: Sumant /Restant dos Nombres**

```
xquery version "1.0";

declare namespace request="http://exist-db.org/xquery/request";
declare namespace util="http://exist-db.org/xquery/util";

declare function local:do-query() as element()
{
  let $n1 := request:request-parameter("n1", ""),
      $n2 := request:request-parameter("n2", ""),
      $op := request:request-parameter("op", "")
  return
    if($n1 = "" or $n2 = "") then
      <p>Please enter two operands.</p>
    else
      let $query := concat($n1, " ", $op, " ", $n2)
      return
        <p>{$query} = {util:eval($query)}</p>
};

<html>
  <body>
    <h1>Enter two numbers</h1>

    <form action="{request:request-uri()}" method="get">
      <table border="0" cellpadding="5">
        <tr>
          <td>First number:</td>
          <td><input name="n1" size="4"/></td>
        </tr>
        <tr>
          <td>Operator:</td>
          <td>
            <select name="op">
              <option name="+">+</option>
              <option name="-">-</option>
            </select>
          </td>
        </tr>
        <tr>
          <td>Second number:</td>
          <td><input name="n2" size="4"/></td>
        </tr>
        <tr>
          <td colspan="2"><input type="submit"/></td>
        </tr>
      </table>
    </form>

    { local:do-query() }
  </body>
</html>
```

### 8.3.5. Transformacions XSL

Exist te una funció per aplicar directament un full d'estils de XSL a un fragment de XML dins d'un XQuery. Això pot ser convenient si els requisits són senzills i no desitgem instal·lar Cocoon en l'aplicació web. La signatura de la funció és així.

```
transform:transform($input as node()?, $stylesheet as
xs:string, $parameters as node()?) as node()?
```

**transform:transform** espera que el node sigui transformat en el primer paràmetre \$input. Si \$input és una seqüència buida la funció retorna immediatament.

El full d'estils XSL serà llegit en la localització especificada en \$stylesheet. Cal que sigui un URI amb path absolut o relatiu. Si especificuem un path relatiu, primerament la funció intentarà trobar un arxiu a emparellar amb la trajectòria. Si això falla, **transform:transform** afegeix el path a la propietat de la base URI actual del context XQuery i revisa.

El full d'estils serà compilat en una plantilla usant l'API estàndard de Java (javax.xml.transform). La plantilla es compartirà entre tots els casos de la funció i es recarregarà solament si està modificada des de la invocació passada.

El paràmetre \$options pot ser usat per a passar paràmetres del full d'estils al processador XSL. Es passen els paràmetres com un fragment XML.:

```
<parameters>
<param name="param1" value="value1"/>
<param name="param2" value="value2"/>
</parameters>
```

Això fixarà el paràmetre *param1* del full d'estils al valor de la cadena *value1*. En el full d'estils el paràmetre pot ser referenciat mitjançant.

```
<xsl:param name="param1"/>
```

En el present cas les transformacions XSL no s'han implementat si no que el que s'ha fet és obtenir codi html directament de la consulta com es pot veure en dels diferents arxius PHP adjunts.

### 8.3.6. Funcions relatives a HTTP

L'exemple anterior llegeix paràmetres de la petició HTTP fent servir la funció **request:request-parameter**. Òbviament, les funcions del namespace `http://exist-db.org/xquery/request` solament es poden fer servir si la consulta és executada pel XQueryGenerator o el XQueryServlet (cal donar un cop d'ull a la [Developer's Guide](#) per a més informació).

**request:request-parameter** compta amb dos arguments: el primer argument denota el nom del paràmetre i la segona el valor prefixat el qual es canviat si no es fa es fixa. La funció retorna retorna una seqüència que conté els valors del paràmetre.

**request:request-uri** retorna el URI de la petició actual. Es pot cridar de la següent manera

```
request:encode-url(request:request-uri())
```

per a codificar la identificació de l'actual sessió de l'URI.

Hi ha també dues funcions per a llegir i fixar els atributs de la sessió. Qualsevol valor XQuery o valor d'objecte Java poden ser emmagatzemats com un atribut de la sessió, per exemple per a dipositar els resultats d'una consulta. Donar un cop d'ull als exemples proporcionats.

La crida a la funció `request:create-session()` crearà una nova sessió HTTP si no s'ha creat cap encara.

Com l'aplicació s'ha fet en mode local aquestes opcions no s'han probat

## 8.3. Lligam Java

Exist suporta crides a mètodes de Java desde dins de XQuery. El mecanisme de lligam segueix una tècnica d'abreujament introduïda per [Saxon](#). La classe on es trobarà la funció externa es identificada pèl namespace URI de la crida a la funció. El namespace URI cal que comenci amb el prefix *java*: seguit pel nom completament qualificat del nom de la classe. Per exemple el codi de l'snippet següent crida l'estàtic mètode *sqrt* de la classe *java.lang.Math*:

**Exemple: Crida a un Mètode Estàtic (Static Method)**

```
declare namespace math="java:java.lang.Math";
math:sqrt(2)
```

Si el nom de la funció conté un guionet, la lletra després del guionet es converteix en majúscula i el guionet és eliminat p.e. *to-string()* cridarà el mètode de Java *toString()*.

Si més d'un mètode de la classe emparella amb el nom i el paràmetre actual, en cas de sobrecàrrega d'operadors, llavors eXisat intenta seleccionar el mètode que millor s'ajusta als tipus que s'han passat com a paràmetre en temps d'execució. El resultat de la crida al mètode es pot assignar a una variable XQuery. Si és possible, serà mapat al tipus corresponent de l'esquema XML. En cas de que no sigui possible l'objecte s'incorporarà al tipus Object.

Els constructors de Java es criden fent servir la funció *new*. Un cop més es selecciona tenint en compte el constructor que emparelli el compte i el tipus de paràmetre. El valor retornat és un nou objecte Java amb el tipus Object.

Els mètodes d'instància són cridats proveint un objecte vàlid de Java com a primer paràmetre. L'objecte de Java té que ser una instància d'una classe donada. Per exemple el següent snippet llista tots els arxius i directoris del directori actual:

**Exemple: Llista de Continguts del Directori Actual**

```
declare namespace file="java:java.io.File";
```

```
<files>
{
  for $f in file:list-files( file:new(".") )
  let $n := file:get-name($f)
  order by $n
  return
    if (file:is-directory($f)) then
      <directory name="{ $n }"/>
    else
      <file name="{ $n }" size="{ file:length($f) }"/>
}
</files>
```

# Capítol 9 Serveis Web

## 9.1. Els serveis Web XML

---

XML està transformant completament la creació i ús de software . Si el web va revolucionar la comunicació entre usuaris i aplicacions XML està revolucionant la comunicació entre aplicacions, i de manera més general la comunicació entre equips ja que ofereix un format de dades universal que permet adaptar o transformar fàcilment l'aplicació.

Els serveis web XML permeten que les aplicacions comparteixin informació i a més puguin invocar funcions d'altres aplicacions independentment de com s'hagin creat aquestes, quin sigui el sistema operatiu, la plataforma en que s'executen o quins dispositius són emprats per tenir accés a elles. Encara que els serveis web són independents entre si, poden vincular-se i formar un grup de col·laboració per a realitzar una tasca determinada.

Els protocols que suporten serveis web es comuniquen normalment pel port 80 i estan basats en HTTP, mètodes GET i PUT. Això fa que es pot accedir a ells de la mateixa manera que ho fem en una pàgina web. La diferència entre una pàgina web i un servei web és que, la pàgina la visita un individu interessat mentre que el servei solament és visitat pels programes que ho requereixen. D'aquesta manera podem dir que el conjunt dels serveis web és una WWW paral·lela en la que els ordinadors parlen entre ells sols a través d'internet.

## 9.2. Els protocols SOAP i XML-RPC

---

Hi ha un conveni generalitzat que ens dona a entendre que els serveis web s'invoquen per internet per mitjà de protocols estàndard basats en XML. Avui en dia hi ha dues grans tendències: XML-RPC i SOAP. Per tant en el moment de programar un servei web cal decidir quin protocol usar ja que un és incompatible amb l'altre.

Tant [SOAP](#) , *Simple Object Access Protocol*, com [XML-RPC](#) són llenguatges estandaritzats pel consorci W3C. Es poden consultar les seves especificacions de manera que coneixerem les regles necessàries per ubicar serveis web XML, integrar-los en aplicacions i establir una comunicació entre ells.

### Diferències entre SOAP i XML-RPC

Breument la diferència entre SOAP i XML-RPC és la seva complexitat. XML-RPC està dissenyat per a ser senzill. SOAP en canvi està creat per a donar un suport complet i al detall de tot tipus de serveis web. L'aprenentatge a XML-RPC permet aconseguir resultats satisfactoris amb poc esforç. A SOAP no passa això però es disposa de més potència. A XML-RPC no es pot escollir el conjunt de caràcters utilitzat. En canvi a SOAP es pot escollir entre US-ASCII, UTF-8 i UTF-16.

Sembla ser que XML-RPC està millor acollit entre la comunitat Unix en canvi SOAP té al seu favor que el projecte Apache disposa d'un mòdul de treball amb SOAP, que es pot usar com una llibreria client per invocar serveis SOAP disponibles en qualsevol lloc o com a eina del costat del servidor per implementar serveis SOAP accessibles.

### Característiques de SOAP.

[SOAP Versió 1.2](#) proporciona la definició de la informació basada en XML que pot ser utilitzada per l'intercanvi d'informació estructurada i de tipus concrets entre punts en un entorn descentralitzat, distribuït.

SOAP és fonamentalment un paradigma per l'intercanvi de missatges en un sol sentit, sense estat, però que les aplicacions poden crear patrons d'interacció més complexos com p.e. petició/resposta, o petició/resposta-múltiple combinant aquests intercanvis d'un sol sentit amb característiques proporcionades pel protocol emprat i informació específica de l'aplicació. SOAP no interfiereix en la semàntica de cap dada específica de l'aplicació que comunica, ni tampoc en assumptes com enrutament de missatges SOAP, transferència de dades fiables,

tallafocs etc. No obstant, SOAP proporciona el marc de treball pel qual la informació d'aplicacions específiques pot comunicar-se de forma extensible. També, SOAP proporciona una descripció completa de les accions que cal que realitzi un node SOAP al rebre un missatge SOAP

Com part de l'enllaç HTTP, presenta dos patrons per l'intercanvi de missatges que estan disponibles per les aplicacions, un utilitza el mètode HTTP POST, i l'altre el HTTP GET. També es proporcionen exemples de com els RPCs, en particular aquells que representen la recuperació d'informació "segura", poden ser representats en intercanvis de missatges SOAP, d'una forma que és compatible amb els principis arquitectònics de la World Wide Web

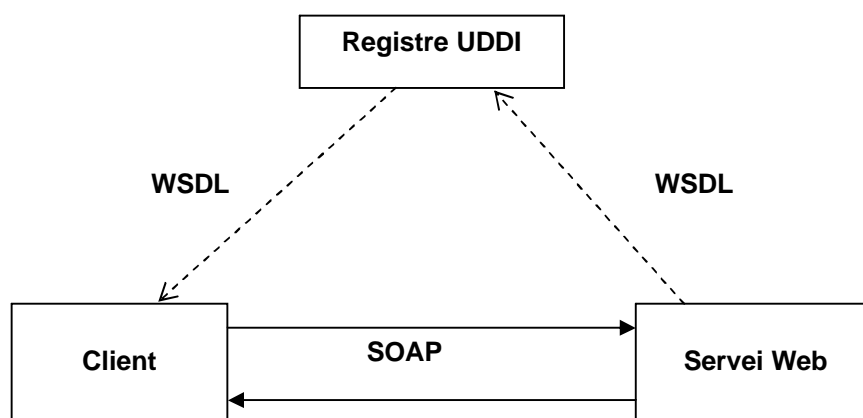
## 9.4. UDDI i els Serveis Web

SOAP a diferència de XML-RPC, inclou una infraestructura al seu voltant. No és un solament un protocol de comunicació entre ordinadors, sinó que a més s'envolta d'especificacions com el WSDL i UDDI. Vegem que signifiquen exactament.

Els serveis Web és un nou paradigma en el desenvolupament de sistemes distribuïts que proveirà d'una plataforma per a totes les transaccions de e-commerce i business-to-business (B2B) a Internet, així com també proporcionar una plataforma universal per a la integració B2B.

La popularitat recent de XML en aquests *sites*, promet una solució per l'intercanvi de dades d'una forma transparent. També l'evolució de protocols com SOAP proporciona una plataforma per l'intercanvi de serveis sobre la xarxa. D'aquesta manera si el mecanisme de comunicació entre plataformes és en XML, i la forma de comunicació és en SOAP, UDDI permet saber amb qui comunicar-se i a on trobar altres negocis.

L'estàndard [UDDI](#) proveeix un mecanisme per a que els negocis se "descriguin" a ells mateixos i els tipus de serveis que proporcionen i llavors després es poden registrar i publicar en un Registre UDDI. Aquests negocis publicats poden ser cercats, consultats o "descoberts" per altres negocis utilitzant missatges amb SOAP. Un cop descoberts els negocis amb qui es poden associar, els negocis poden utilitzar aquest mecanisme per "integrar" els seus serveis en conjunció amb els seus socis i proveir els serveis als seus clients.



### Pila d'interoperabilitat dels serveis web

Un servei web és un servei, amb una interfície definida i coneguda, al que es pot accedir a través d'internet. Igual que una pàgina web està definida per un URL (Uniform Resource Locator), un servei web està definit per un URI (Uniform Resource Identification) i per la seva interfície a través de la que es pot accedir a ell. D'aquesta manera les aplicacions es converteixen en clients que integren serveis web procedents de diferents proveïdors.

Pila d'interoperabilitat dels serveis web	UDDI (Universal Description, Discovery and Integration)
	SOAP (Simple Object Access Protocol)
	XML (eXtensible Markup Language)
	Protocols comuns d'internet

Descripció detallada dels elements de la pila:

- **Protocols Comuns d'Internet** com el HTTP, SMTP, FTP i d'altres. Estableix el marc de referència bàsic pels serveis Web, encara que els serveis web no estan lligats a cap protocol en particular.
- **XML** s'ha convertit en l'estàndard universal per l'intercanvi de dades i de les seves estructures i estableix la base dels serveis web.
- **SOAP** proporciona un mecanisme senzill i lleuger per l'intercanvi d'informació estructurada i textual amb XML en un ambient descentralitzat i distribuït.
- **UDDI** defineix una forma de publicar i descobrir informació referent als serveis Web. Un núvol de registres UDDI, proporciona informació de com accedir als serveis Web.

## WSDL

El llenguatge de descripció de serveis Web [WSDL](#) és un llenguatge XML que conté informació referent a la interfície, semàntica i administració d'una crida a un servei web.

Un cop s'ha desenvolupat un servei Web, es publica la seva descripció i es construeix un lligam o apuntador en un depòsit UDDI per a que els usuaris potencials el puguin utilitzar. Quan algú pensa utilitzar aquest servei Web, sol·liciten l'arxiu WSDL per a conèixer la ubicació del servei, crida de funcions i com accedir al propi servei web. Llavors utilitzen la informació en l'arxiu WSDL per a construir una petició SOAP i enviar-la cap al proveïdor del servei.

Degut a que l'aplicació s'ha fet funcionar en local això està inclòs a l'arxiu `exist_soap.php` de manera que creant una variable `wsdl` que comunica amb `http://localhost:8888/exist/services/Query?wsdl` pren la seva informació i la utilitza per a crear una instància d'un SoapClient, és a dir, construeix la petició SOAP.

El WSDL és l'equivalent d'un resum en XML – describint els serveis Web, on s'ubiquen, i com es poden invocar. Per aquesta raó s'ha tingut que carregar l'arxiu `Query.xml` dins del subdirectori `$CATALINA_HOME/webapps/exist/services`. En ell es descriuen en XML les definicions dels tipus, missatges, tipus de port, operacions i serveis WSDL.

La especificació UDDI, juntament amb Extensible Markup Language (XML), Simple Object Access Protocol (SOAP) i Web Services Description Language (WSDL), estan guanyant un ampli suport dins del marc de treball dels serveis Web. Avui en dia prop de 7.000 empreses s'han registrat en el directori UDDI.

# Capítol 10 Aplicació per a la gestió d'un Centre d'Estudis

## 10.1 Requisits previs.

L'aplicació gestionarà les dades dels alumnes matriculats. Aquests es deuen organitzar per tipus d'estudis, cursos i anys acadèmics. Es poden emmagatzemar els temes d'interès de l'alumne, els seus amics, i el nom de les organitzacions a la que pertany. En aquest cas aquestes organitzacions seran les empreses on presta actualment o ha prestat en anterioritat els seus serveis com alumne en pràctiques.

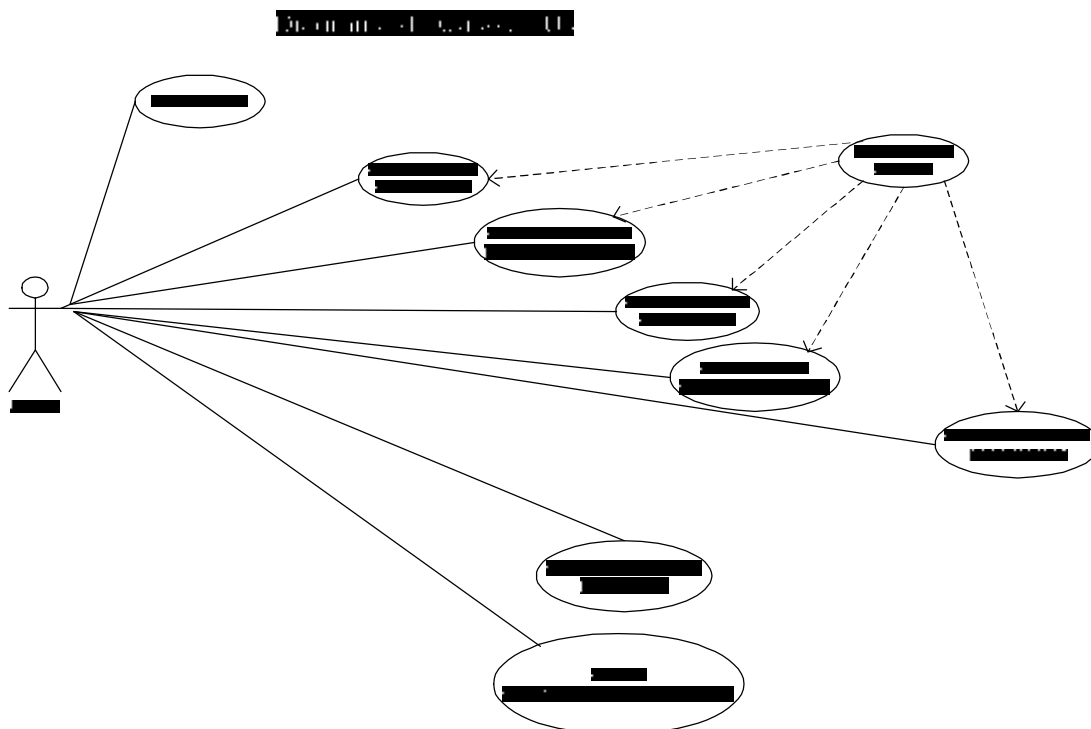
És clar que per a discriminar les operacions a realitzar en la base de dades cal un procés d'autenticació per part de l'usuari, que si s'ha superat, restringirà l'actuació d'aquest en la bdd.

- Autenticació

El tipus de consultes a realitzar són

- Consulta de la ficha personal de l'alumne partir d'el nom i cognoms i del curs en que està matriculat.aquesta elecció es farà a partir d'una elecció prèvia d'una llista ordenada pel cognom dels alumnes per unes determinades condicions establerta i sel-leccionat en primera instància
- Consulta d'amics d'un alumne
- Consulta dels alumnes interessats per un tema.
- Consulta estadística de matriculació, número d'alumnes per a cada grup.
- Consulta de l'historial professional d'un alumne
- Consulta d'alumnes que treballen o han treballat en una organització determinada

## Diagrama de Casos d'Ús.



## 10.2 Anàlisi.

## Cas d'Ús nº 1: AUTENTICACIÓ

**Resum de la funcionalitat:** Permet a un usuari autenticar-se degut a que realitza una consulta a la BBDD tant de idUser com password

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

**Casos d'ús relacionats:**

**Precondició:** cap

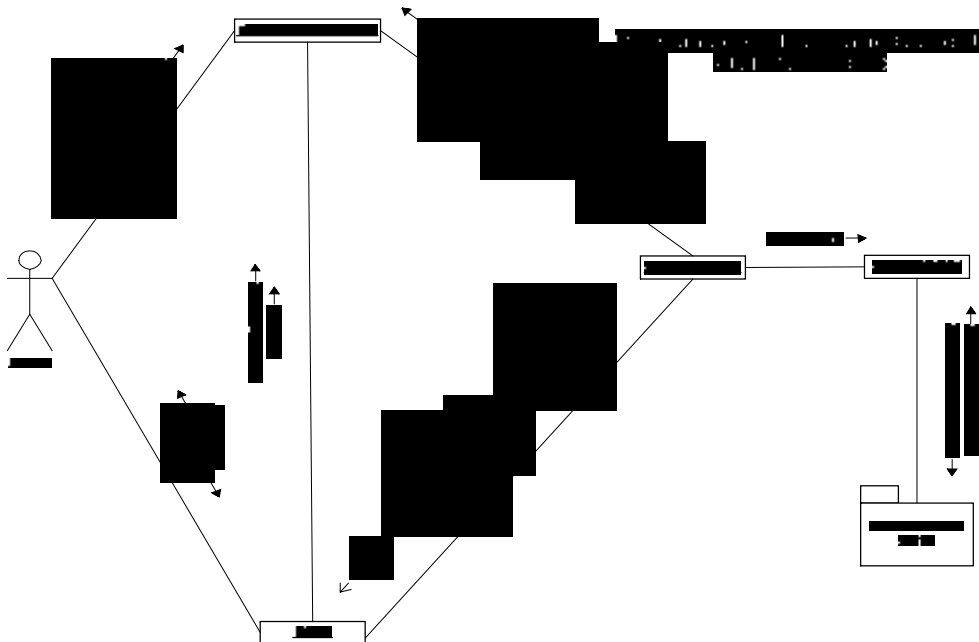
**Postcondició:** l'usuari ha estat validat i per tant s'inicia una sessió en l'aplicació

**Procés normal principal:**

1. El sistema demana codi i clau d'usuari, l'usuari l'introdueix i prem el botó acceptar
2. El sistema ha iniciat una sessió de treball
3. El sistema mostra la l'index de l'aplicació segura de la qual pot escollir qualsevol opció
4. Si l'usuari escull sortir de l'aplicació el sistema torna a mostrar la pantalla d'autenticació

**Alternatives de procés i excepcions:**

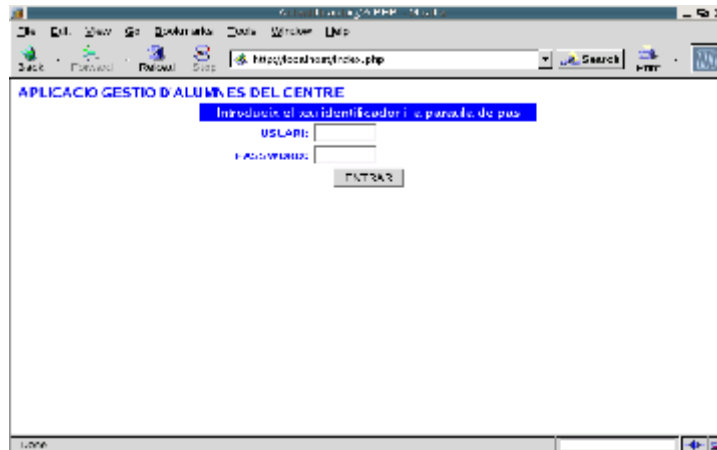
1. El codi d'usuari o la clau no existeix o no corresponen:
  - 1.1. El sistema treu un missatge d'error dins de la pantalla d'identificació d'usuari



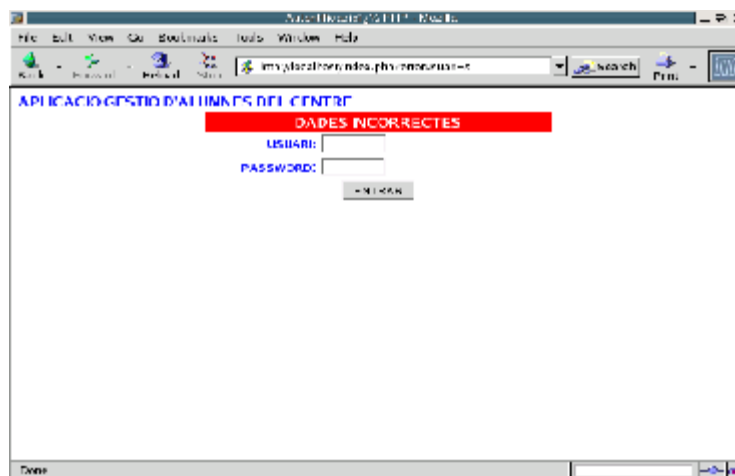
**Disseny de la interfície d'usuari PantallaAutenticació:**

Caldrà incloure com a usuari "admin" i com password deixar el camp en blanc.



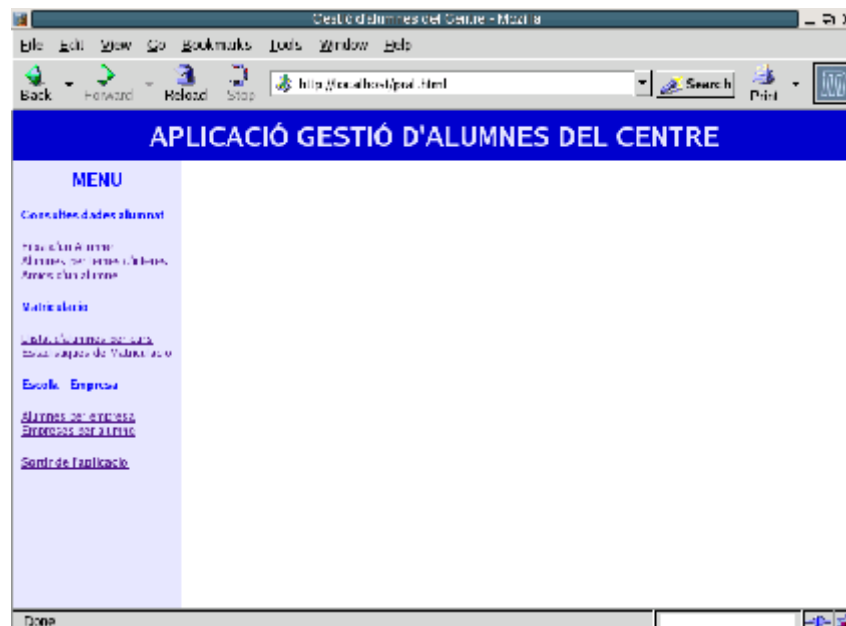


en cas de no incloure com a usuari "admin" i password en blanc apareixerà la mateixa pantalla però amb el següent aspecte:



### Disseny de la interfície d'usuari Menu

El menú apareixerà en les posteriors interfícies d'usuari en un frame en el costat esquerra i així sempre serà disponible. En cas de voler sortir es premerà l'hipervincle corresponent i apareixerà la pantalla inicial d'autenticació



## Cas d'Ús nº 2: CONSULTA FITXA D'UN ALUMNE

**Resum de la funcionalitat:** Permet consultar les dades personals d'un alumne

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

**Casos d'ús relacionats:**

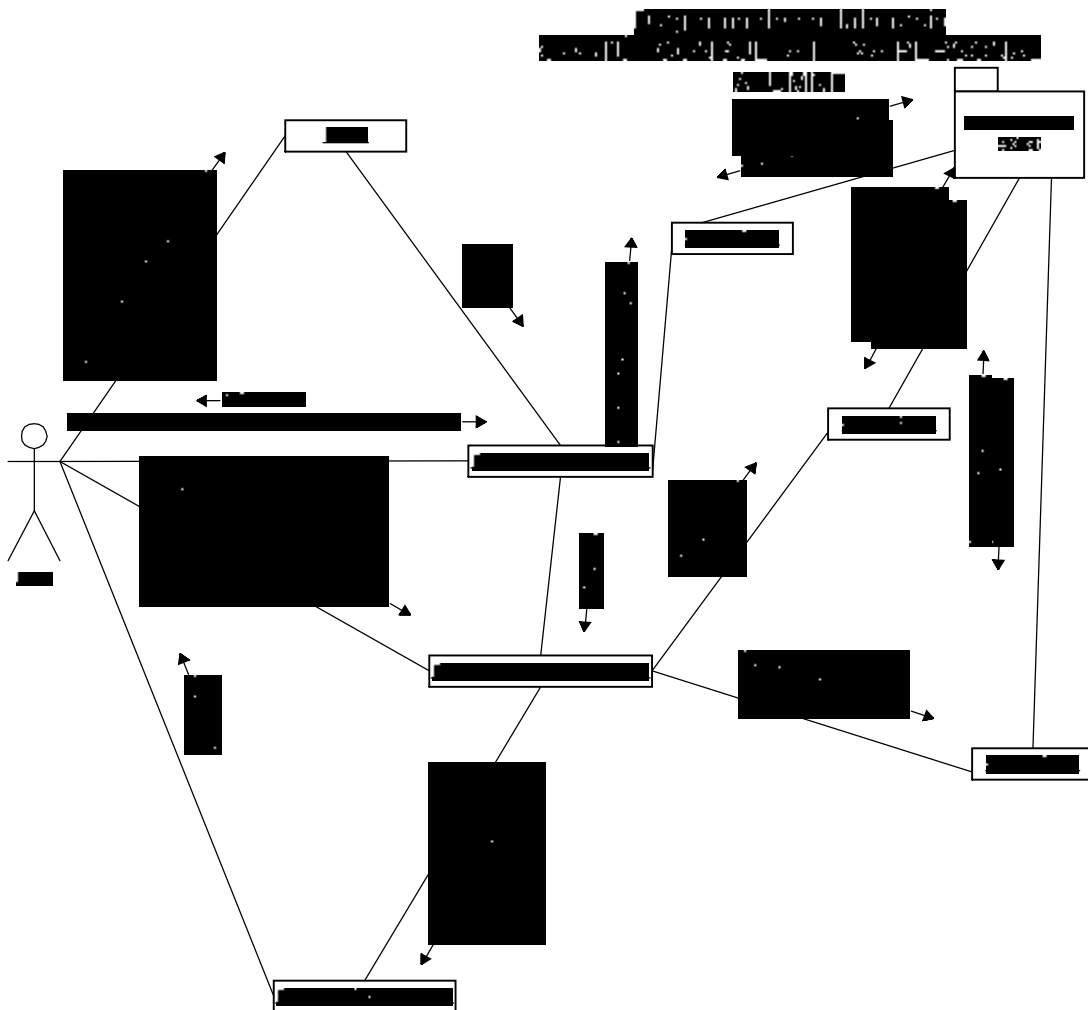
**Precondició:** l'usuari cal que s'hagi autenticat

**Postcondició:**

**Procés normal principal:**

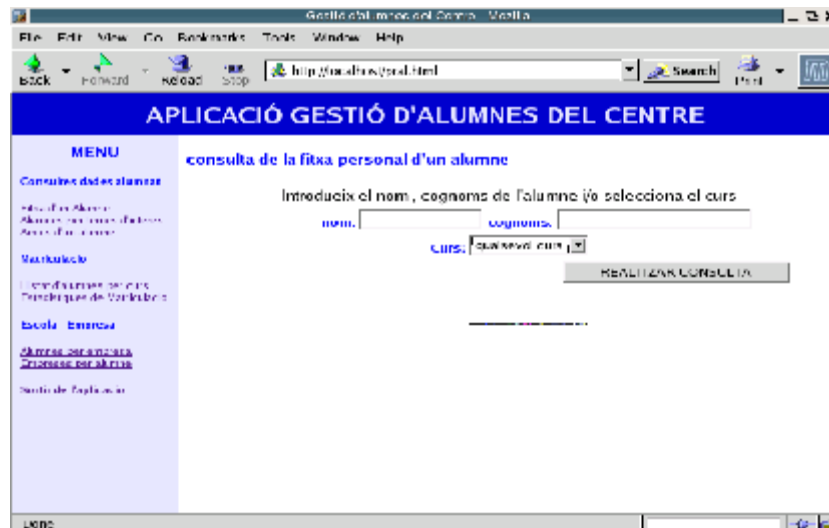
1. El sistema demana nom cognoms i grup al que pertany l'alumne, l'usuari l'introdueix i prem el botó "Realitzar Consulta"
2. El sistema mostra una relació d'alumnes que compleixen els requisits
3. L'usuari pot escollir a un de la llista
4. El sistema mostra la fitxa de l'alumne seleccionat

**Alternatives de procés i excepcions:**



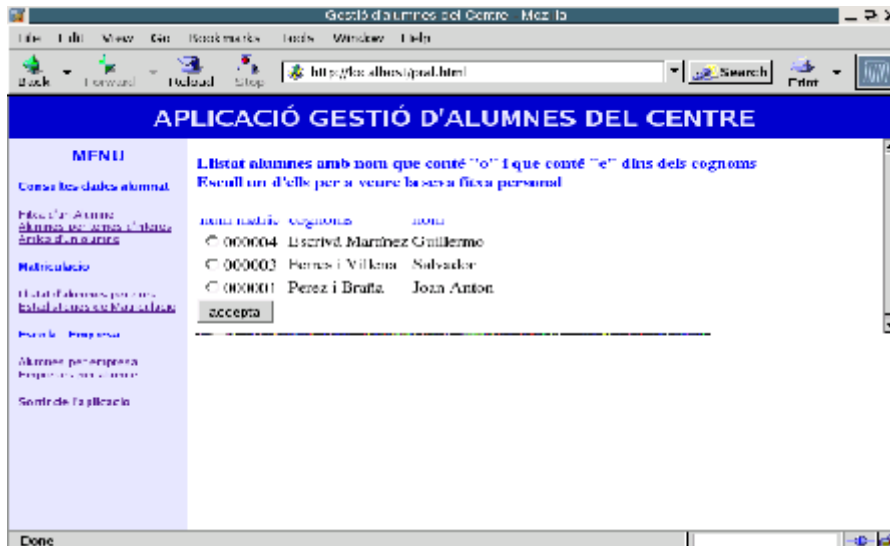
### Disseny de la interfície d'usuari Pantalla\_intro\_alumne:

En el frame 1 apareixen 3 quadres de text on s'introdueixen el nom i els cognoms de l'alumne i s'escullirà del tercer el curs a que pertany. Si el nom complet de l'alumne o el cognom no es coneixen es pot posar part del contingut. Es pot escollir també l'opció qualsevol curs si no es coneix el curs al que pertany



### Disseny de la interfície d'usuari Pantalla\_seleccio\_alumnes:

En el frame 1 apareix una llista d'alumnes que correspon als criteris de cerca introduïts anteriorment, dels quals caldrà escollir un i en prémer el botó acceptar apareixerà en el frame 2 la fitxa de l'alumne



### Disseny de la interfície d'usuari Pantalla\_fitxa-alumne:

En el frame 2 apareix la foto de l'alumne i algunes característiques d'ell com nom, cognoms, número de matriculació, gènere, telèfon, web personal, i els temes d'interès



**Cas d'Ús nº 3: CONSULTA D'ALUMNES PER TEMES D'INTERÈS**

**Resum de la funcionalitat:** Permet consultar els alumnes que tenen interès per un o dos temes, amb opció and /or.

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

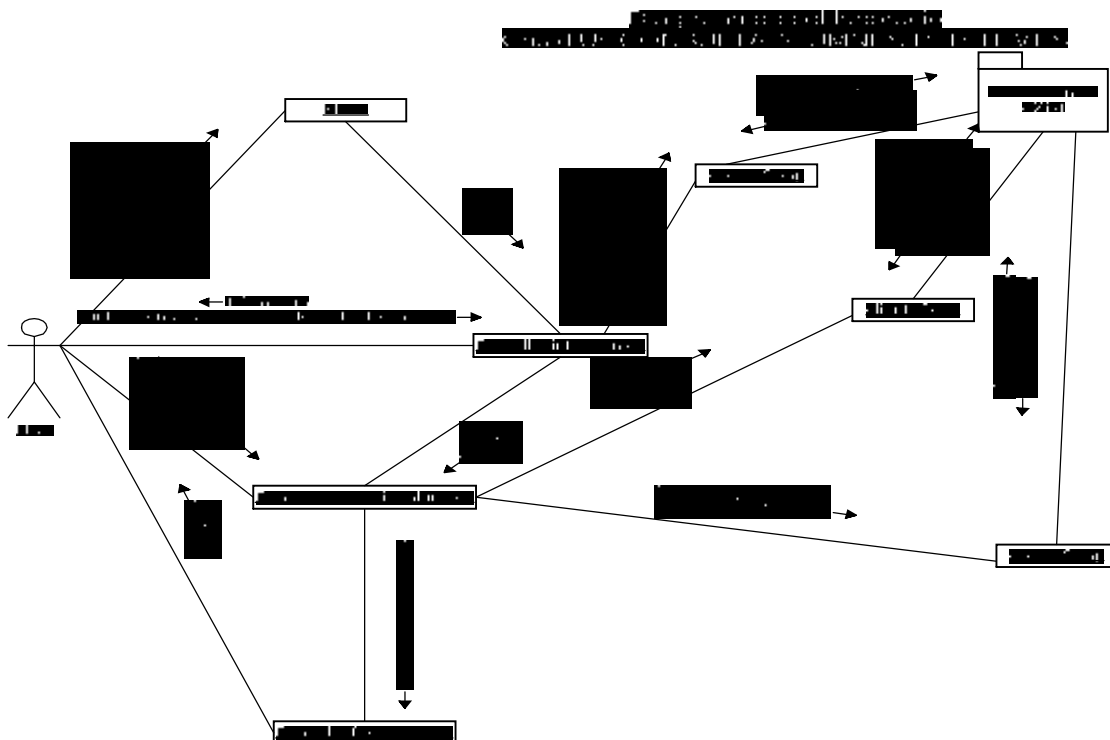
**Casos d'ús relacionats:**

**Precondició:** l'usuari cal que s'hagi autenticat

**Postcondició:**

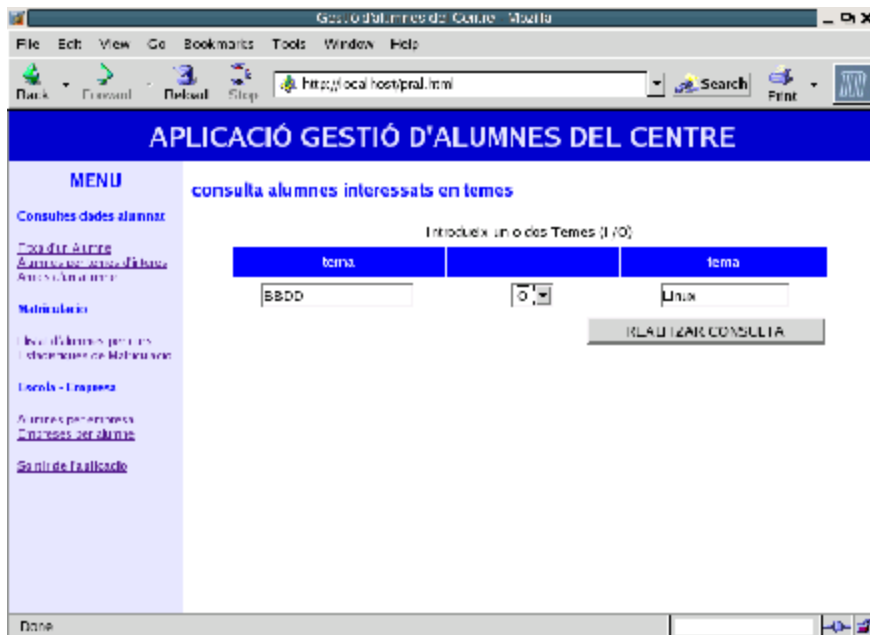
**Procés normal principal:**

1. El sistema demana els temes objecte de cerca i un operador lògic, "and" o "or", l'usuari els introdueix i prem el botó "Realitzar Consulta"
2. El sistema mostra una relació d'alumnes que compleixen els requisits
3. L'usuari pot escollir a un de la llista
4. El sistema mostra la fitxa de l'alumne seleccionat



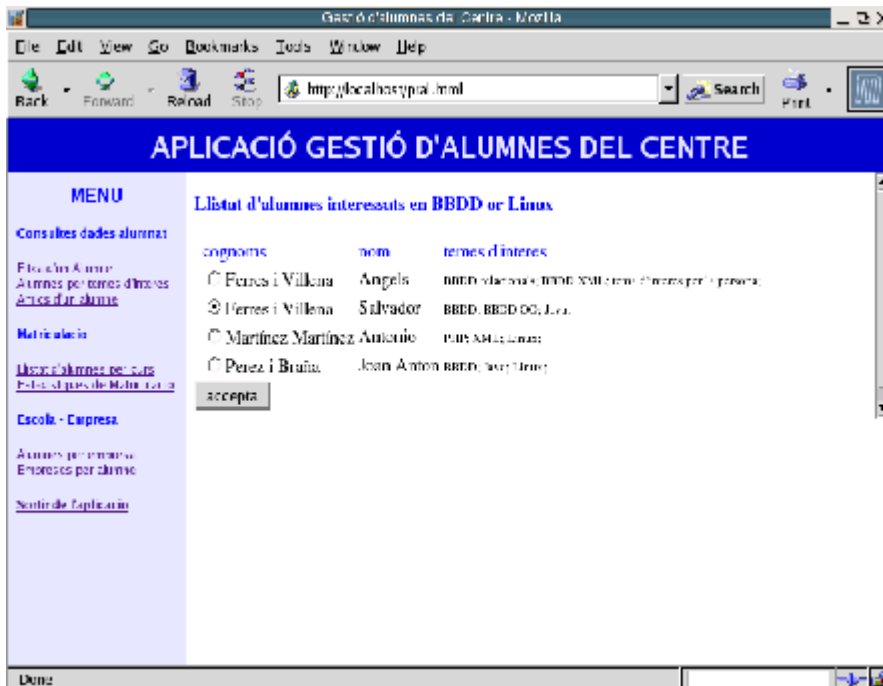
**Disseny de la interfície d'usuari Pantalla\_intro\_temes:**

S'introdueixen els temes a cercar de manera que el resultat correpondrà a l'operació lògica (and o or, aquesta per defecte) dels temes que continguin el que s'ha introduït en la caixa de text.



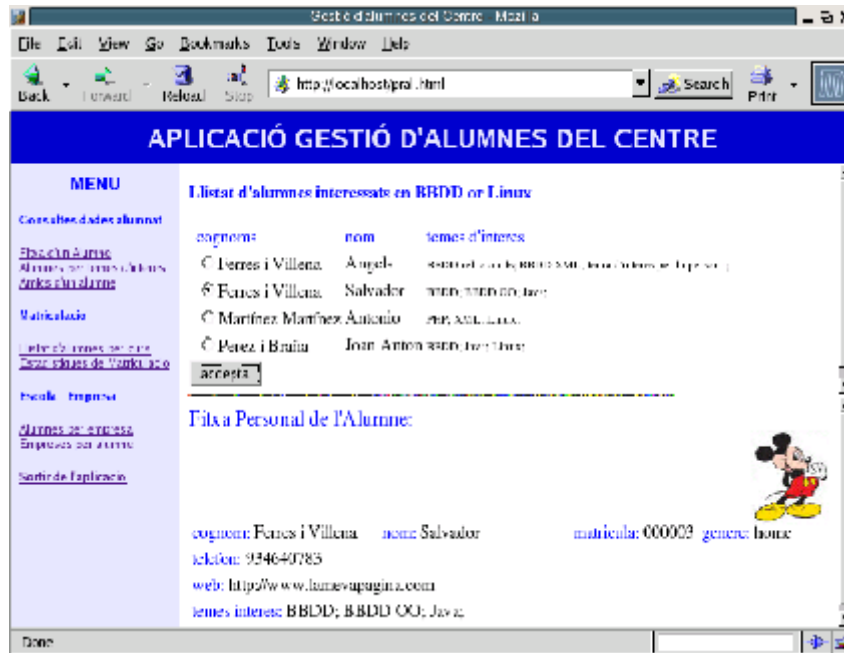
**Disseny de la interfície d'usuari Pantalla\_seleccio\_alumnes:**

En el frame 1 apareixerà una llista d'alumnes que compleixin les condicions de la cerca anterior i d'aquests s'escollirà un de manera que el resultat apareixerà en el frame 2 .



**Disseny de la interfície d'usuari Pantalla\_fitxa-alumne:**

En el frame 2 apareix la foto de l'alumne i algunes característiques d'ell com nom, cognoms, número de matriculació, gènere, telèfon, web personal, i els temes d'interès



**Cas d'Ús nº 4: CONSULTA DELS AMICS D'UN ALUMNE**

**Resum de la funcionalitat:** Permet consultar les dades personals d'un alumne amic d'un altre

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

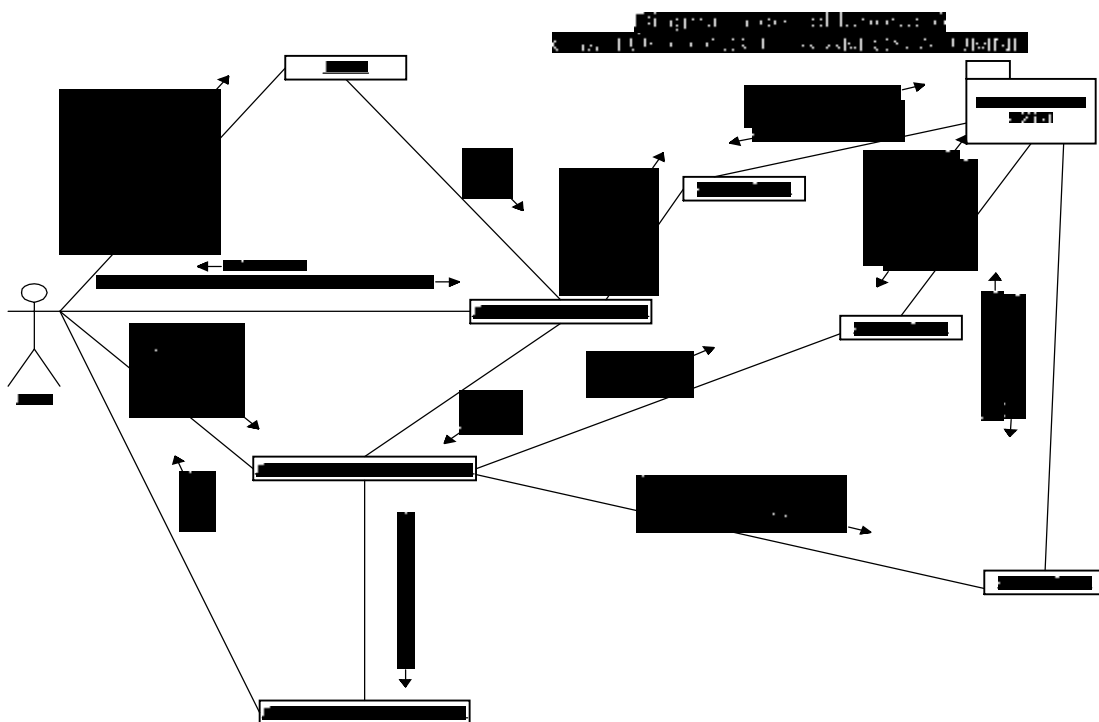
**Casos d'ús relacionats:**

**Precondició:** l'usuari cal que s'hagi autenticat

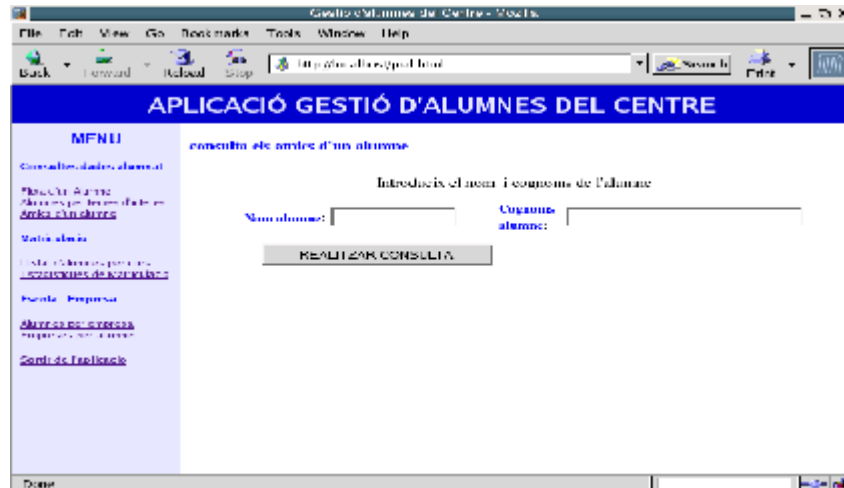
**Postcondició:**

**Procés normal principal:**

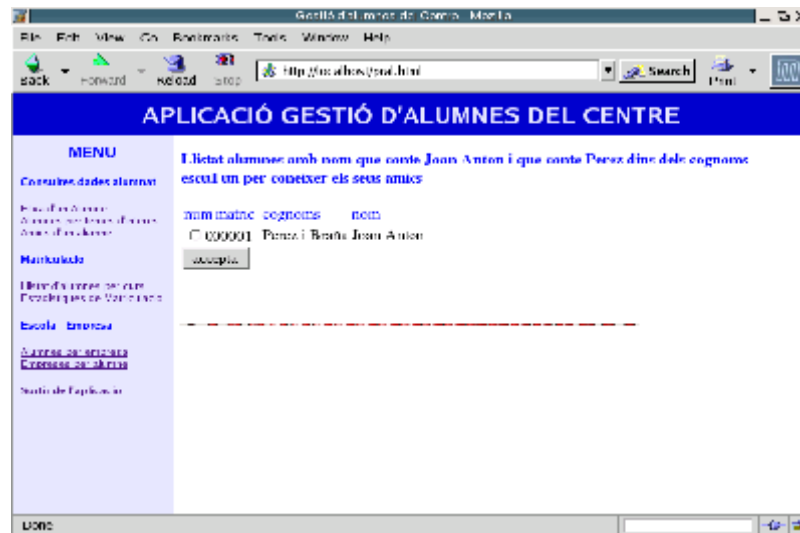
1. El sistema demana nom i cognoms d'un alumne, l'usuari l'introdueix i prem el botó "Realitzar Consulta"
2. El sistema mostra una relació d'alumnes que compleixen amb els requisits de cerca
3. L'usuari pot escollir a un de la llista
4. El sistema mostra la fitxa de l'alumne seleccionat i els seus amics



### Disseny de la interfície d'usuari Pantalla\_cerca\_amics



### Disseny de la interfície d'usuari Pantalla\_seleccio\_alumnes



### Disseny de la interfície d'usuari Pantalla\_alumne\_i\_amics



### Cas d'ús nº 5: OBTENIR LLISTATS DE MATRICULACIÓ

**Resum de la funcionalitat:** Permet consultar les dades personals d'un alumne a partir del llistat d'alumnes d'un grup

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

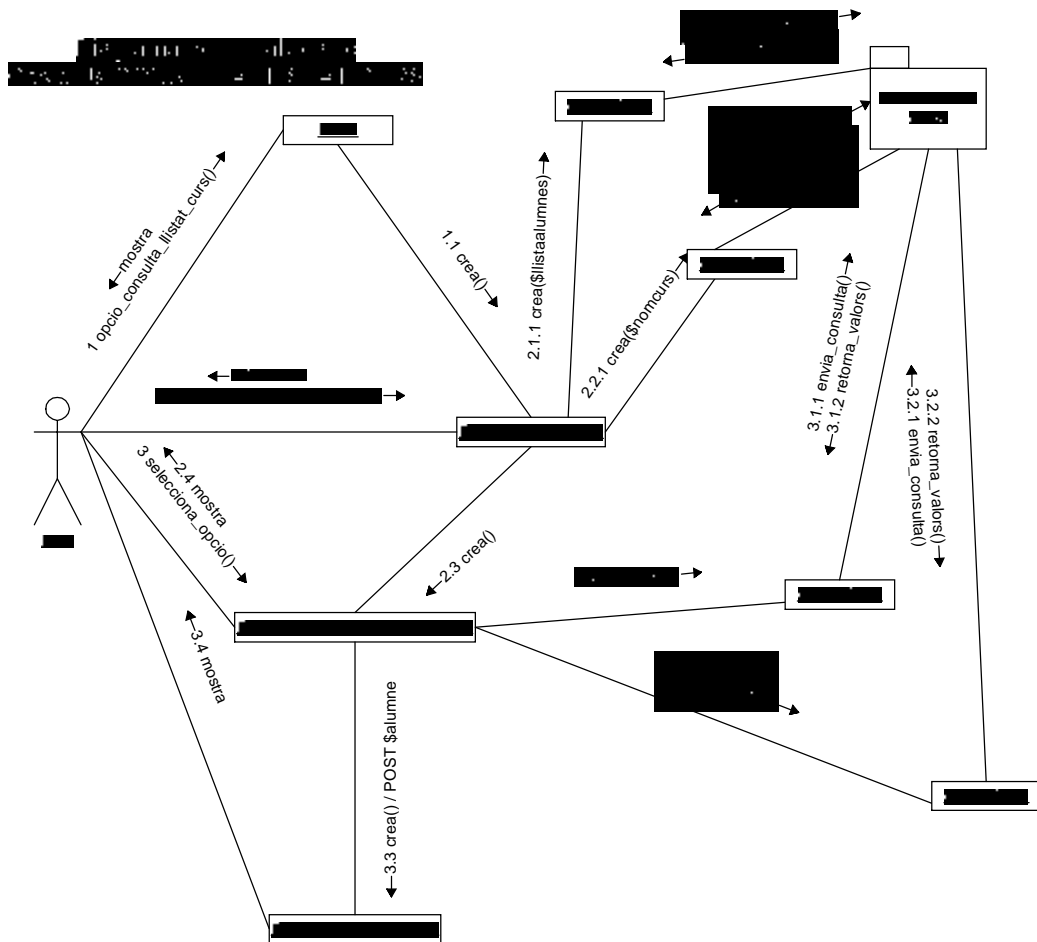
**Casos d'ús relacionats:**

**Precondició:** l'usuari cal que s'hagi autenticat

**Postcondició:**

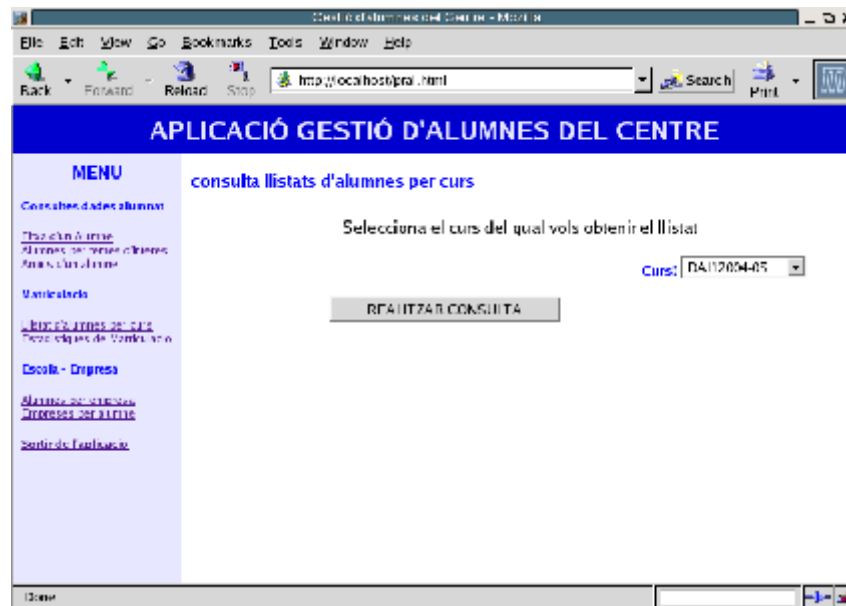
**Procés normal principal:**

1. El sistema demana el grup del qual es mostrarà un llistat de tots els seus membres quan s'hagi clicat el botó "Realitzar Consulta"
2. El sistema mostra una relació d'alumnes que compleixen els requisits
3. L'usuari pot escollir a un de la llista
4. El sistema mostra la fitxa de l'alumne seleccionat



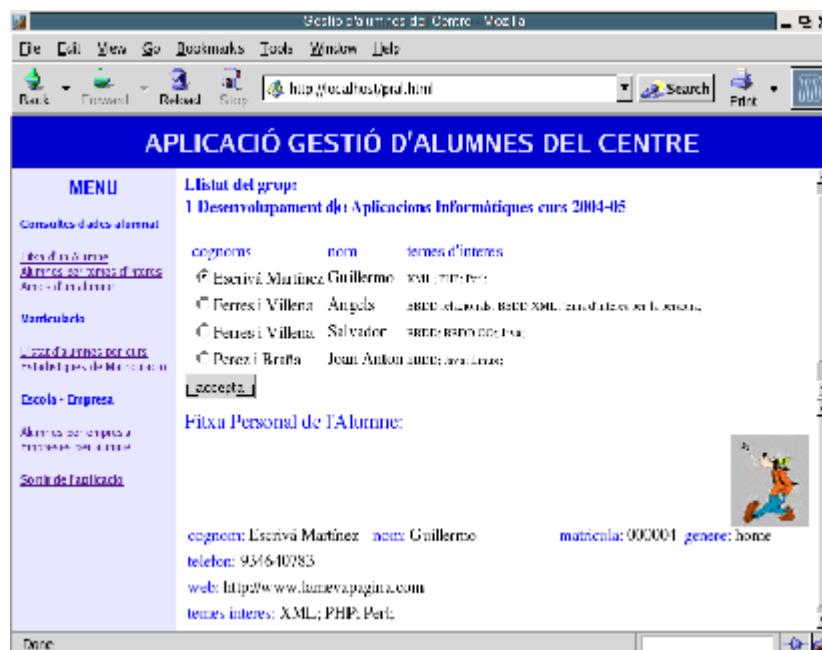
Disseny de la interfície d'usuari Pantalla\_llistat\_curs





### Disseny de la interfície d'usuari Pantalla\_seleccio\_alumnes\_curs:

Apareix en el frame 1 mentre en el frame 2 en haver escollit un d'ells apareix la fitxa personal de l'alumne.



### Cas d'Ús nº 6: CONSULTA D'ALUMNES PER EMPRESA

**Resum de la funcionalitat:** Permet consultar les dades personals d'un alumne que ha tingut una relació amb una empresa determinada

**Paper dins del treball de l'usuari:** bàsic per a tots els actors

**Actors:** Usuari

**Casos d'ús relacionats:**

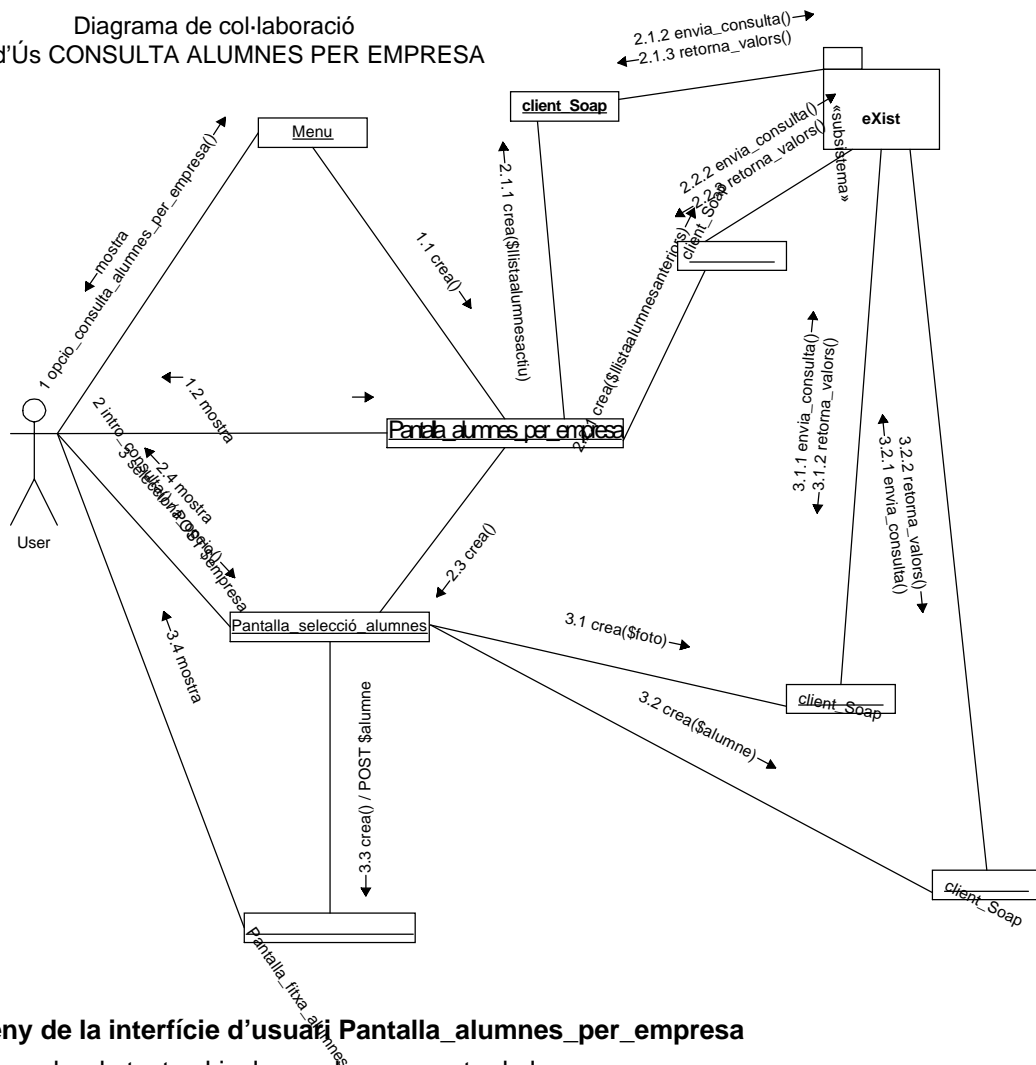
**Precondició:** l'usuari cal que s'hagi autenticat

**Postcondició:**

**Procés normal principal:**

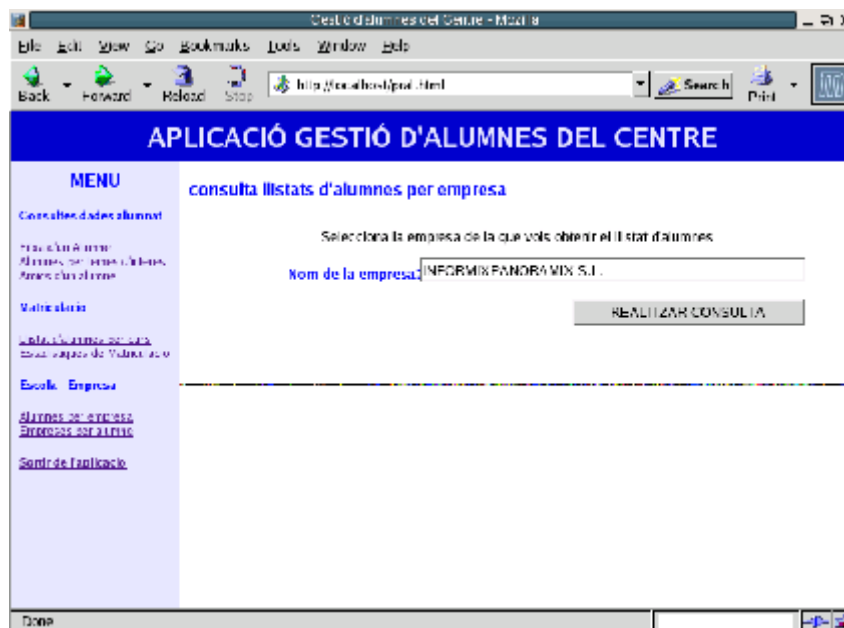
1. El sistema demana nom d'una empresa l'introdueix i prem el botó "Realitzar Consulta"
2. El sistema mostra una relació d'alumnes que han tingut una relació amb aquesta empresa
3. L'usuari pot escollir a un de la llista
4. El sistema mostra la fitxa de l'alumne seleccionat

Diagrama de col·laboració  
Cas d'ús CONSULTA ALUMNES PER EMPRESA



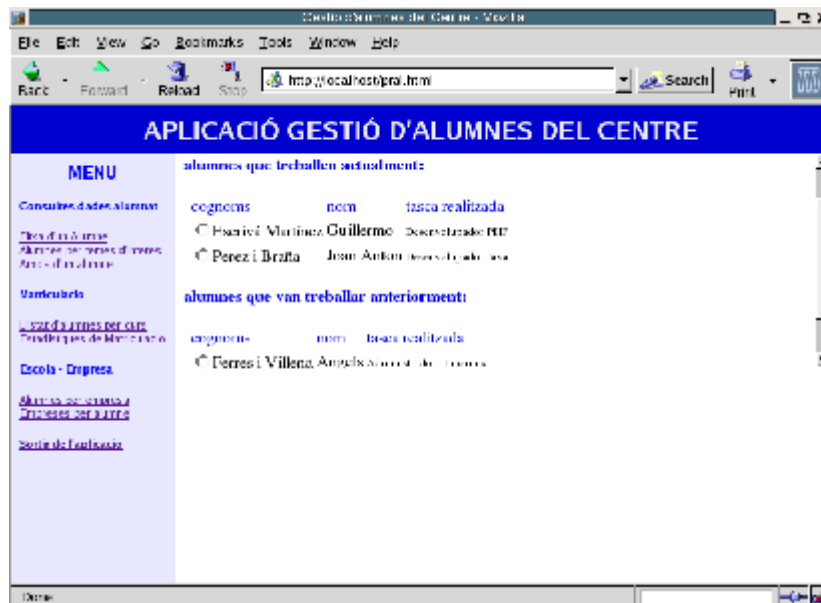
Disseny de la interfície d'usuari Pantalla\_alumnes\_per\_empresa

En el quadre de text cal incloure el nom exacte de la empresa

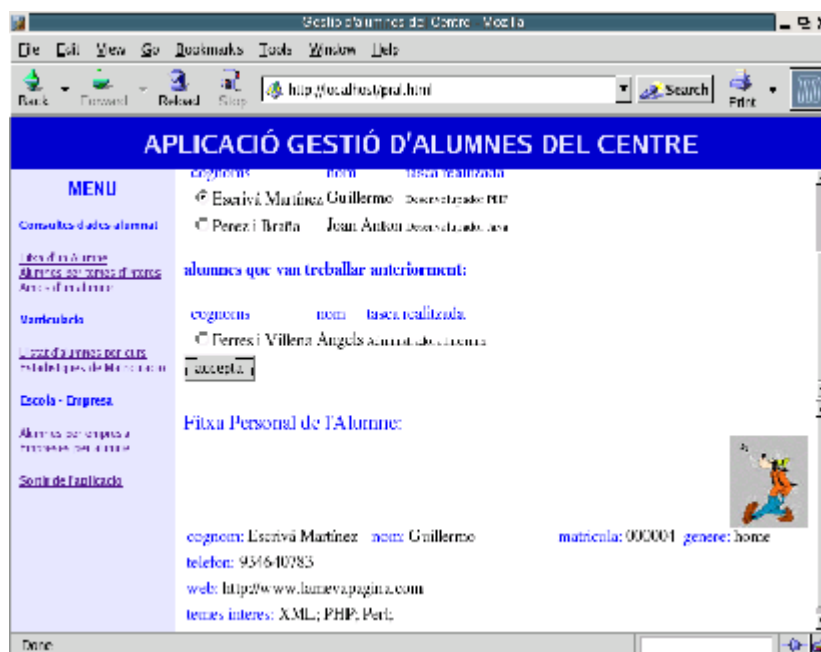


Disseny de la interfície d'usuari Pantalla\_seleccio\_alumnes\_empresa

Apareix una relació dels alumnes que treballen actualment i dels que ho han fet amb anterioritat. En escollir un d'ells apareixerà la seva fitxa en el frame 2



### Disseny de la interfície d'usuari Pantalla\_fitxa\_alumne



### Cas d'Ús nº 7: OBTENIR ESTADÍSTIQUES DE MATRICULACIÓ

**Resum de la funcionalitat:** Permet a un usuari consultar el nombre d'alumnes matriculats a cada curs

**Paper dins del treball de l'usuari:** esporàdic per a User i Admin

**Actors:** User

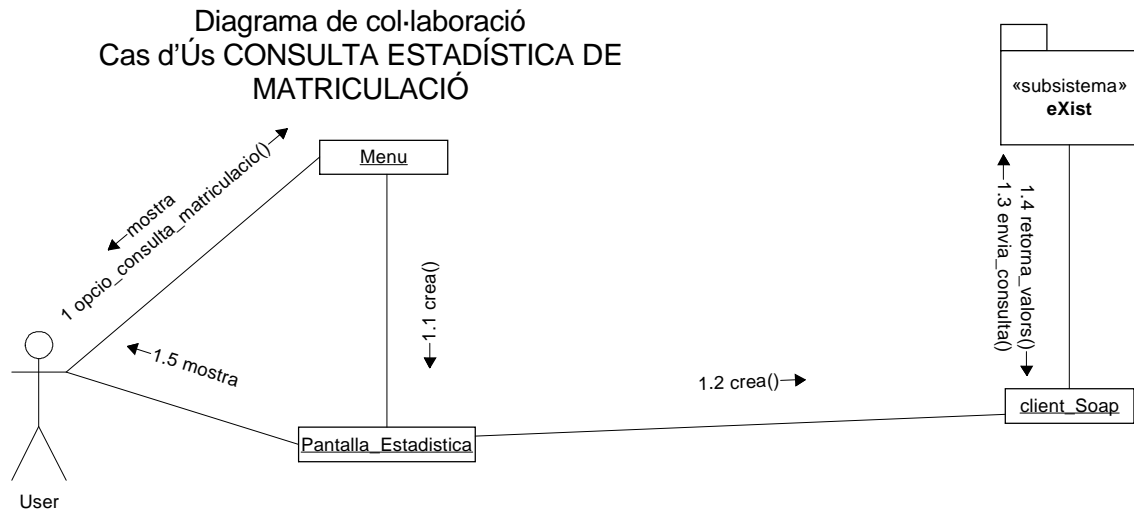
**Casos d'ús relacionats:** Autenticació

**Precondició:** L'Usuari cal que s'hagi autenticat i s'hagi iniciat una sessió de treball

**Postcondició:**

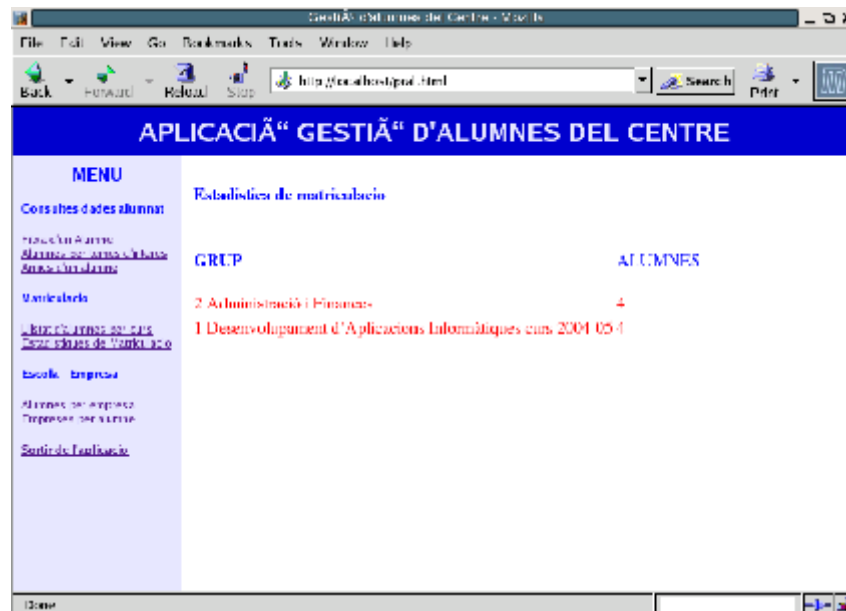
**Procés normal principal:**

1. En escollir l'opció corresponent del menú el sistema mostra un formulari on per a cada curs apareix el nombre d'alumnes matriculats



### Disseny de la interfície d'usuari Pantalla\_estadistica

Mostra per a cada grup el nombre d'alumnes matriculats



### Cas d'Ús nº 8: CONSULTA D'EMPRESES PER ALUMNE

**Resum de la funcionalitat:** Permet a un usuari consultar l'empresa actual i les anteriors on ha treballat un alumne.

**Paper dins del treball de l'usuari:** bàsic per a User

**Actors:** User

**Casos d'ús relacionats:** Autenticació

**Precondició:** L'Usuari cal que s'hagi autenticat i s'hagi iniciat una sessió de treball

**Postcondició:**

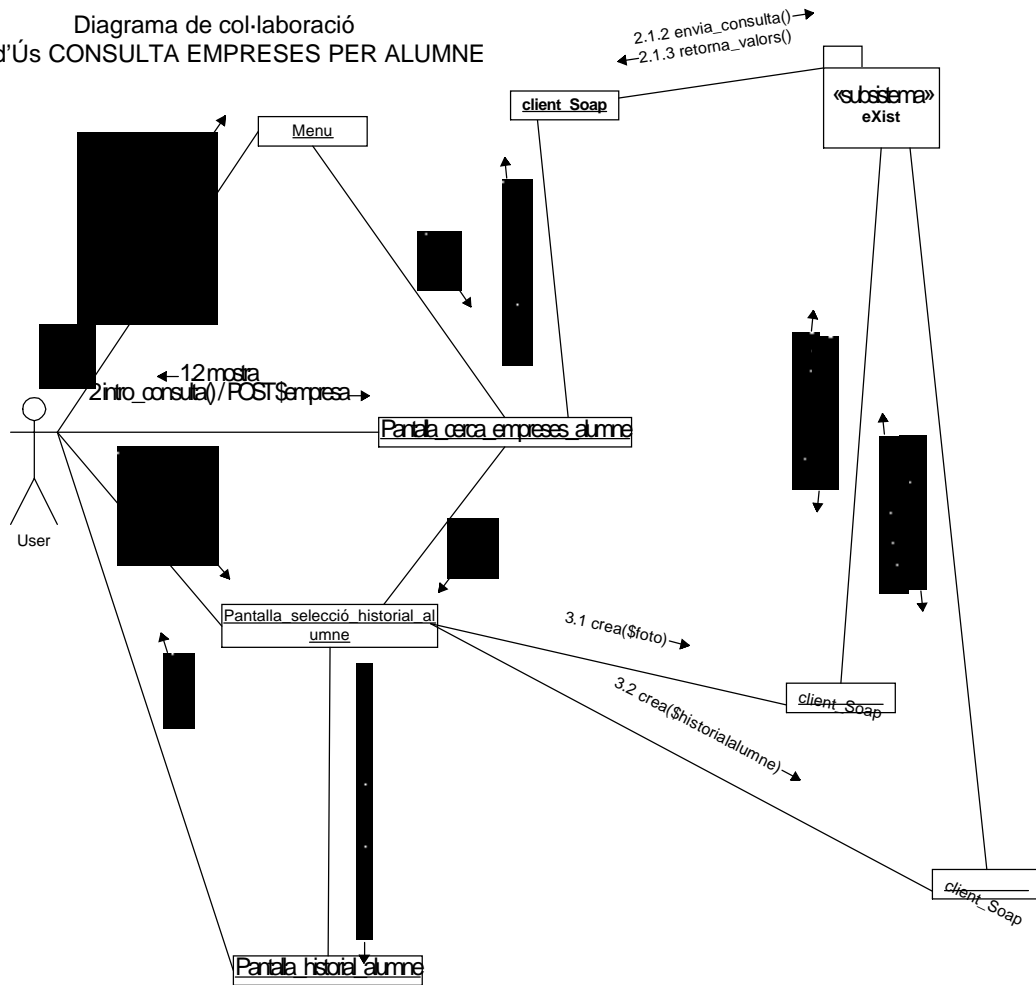
**Procés normal principal:**

1. El sistema mostra un formulari on l'usuari introduirà les dades per a identificar l'alumne, és a dir en nom i cognoms
2. L'usuari prem el botó acceptar
3. El sistema mostra els alumnes que corresponen am els criteris de cerca
4. l'usuari escull un d'ells
5. el sistema mostra dades de l'alumne, les dades de l'empresa actual i les de les feines anteriors

**Alternatives de procés i excepcions:**

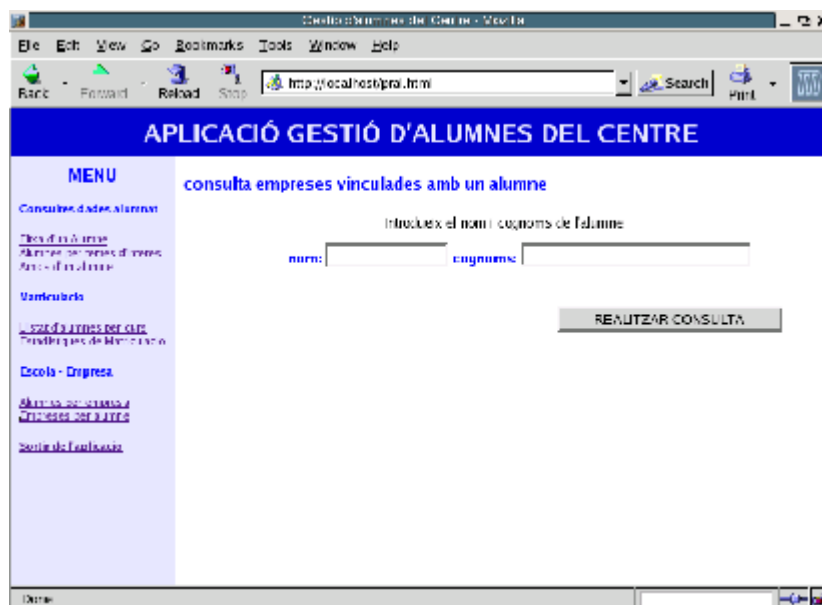
En cas de que no s'hagi pogut trobar l'alumne o no té cap historial professional apareix un missatge aclaridor del fet

Diagrama de col·laboració  
Cas d'Ús CONSULTA EMPRESSES PER ALUMNE



**Disseny de la interfície d'usuari Pantalla\_cerca\_empreses\_alumne(frame 1)**

Permet introduir el nom i cognoms de l'alumne i en premer la tecla realitzar consulta apareix el resultat en el frame 2

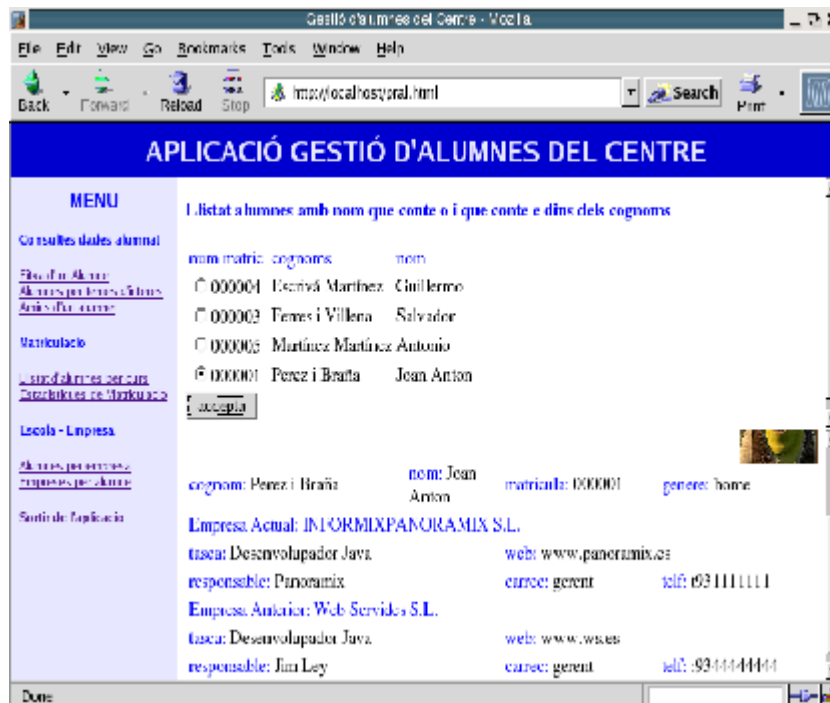


**Disseny de la interfície d'usuari Pantalla\_seleccio\_historial\_alumne(frame 1)**

Permet introduir el nom i cognoms de l'alumne i en premer la tecla realitzar consulta apareix el resultat en el frame 2

### Disseny de la interfície d'usuari Pantalla\_historial\_alumne(frame 2)

Apareixen dades de l'alumne, la seva foto i per a cada empresa, ja sigui actual o en la que ha estat treballant la tasca realitzada, el nom de la empresa, la seva web i el nom del responsable, càrrec i telèfon en el frame 2



## 11.3 Disseny.

Sembla ser que l'aplicació podria córrer sobre una interfície gràfica d'usuari tipus HTML fent crides a un servidor web (servidor de l'aplicació). Serà el servidor de l'aplicació qui es comunicarà amb el servidor BD:XML

Cada cas d'ús descrit en la fase d'anàlisi s'implementarà basant-se en una consulta a partir de les regles i de les restriccions que permet XQuery, fins i tot la generació de l'aspecte del document de sortida es farà mitjançant XQuery.

El disseny de les interfícies es pot observar en cada cas d'ús.

Per a minimitzar problemes de redundància de dades i d'integritat i d'inconsistència en la base de dades s'ha optat per que aquestes estiguin en dos tipus de documents. Un primer tipus de document que descriurà les característiques de l'alumne i un segon tipus de document que descriurà un grup, dins d'un tipus d'estudi i corresponent a un període acadèmic.

Pel que fa al modelatge dels alumnes s'opta per fer un únic document per a cadascú i de manera que aquest s'emmagatzemarà en una única col·lecció. El nom del document pot correspondre al valor de qualsevol atribut que pugui ser clau. Una opció és utilitzar l'atribut de l'element Person que es troba més a l'arrel del document i que descriu a la persona que és alumne. Aquest atribut que anomenarem rdf:nodeID correspondrà amb el número de l'expedient que també és únic per a cada alumne. Posteriorment es descriu el nom i cognoms, nick, gènere (home o dona). S'observa la possibilitat d'incorporar posteriorment la imatge de la persona en l'aplicació i dins de la fitxa de l'alumne poder consultar el telèfon i la seva pàgina web. Posteriorment es reflexen els temes d'interès de la persona i les dades de la feina actual i les de les anteriors.

Un document FOAF d'un sol alumne tindria el següent esquema:

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>
```

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

  <foaf:Person rdf:nodeID="000001">
    <foaf:givenname>Joan Anton</foaf:givenname>
    <foaf:surname>Perez i Braña</foaf:surname>
    <foaf:nick>Joan</foaf:nick>
    <foaf:gender>home</foaf:gender>

  <!-- Aquí faig la descripció de la imatge de la persona....-->
    <foaf:depiction>
      <foaf:Image rdf:resource="http://www.elmeusevidor.com/joan/imatgeJoan"/>
    </foaf:depiction>

  <!--Altres dades de la persona....-->
    <foaf:homepage rdf:resource="http://www.lamevapagina.com"/>
    <foaf:phone rdf:resource="934640783"/>
    <foaf:mbox_sha1sum>75e7702347baf6222e007f63e890b716a7d4a</foaf:mbox_sha1sum>
    <foaf:schoolHomepage rdf:resource="http://www.LaPaginaDeLaEscola.com"/>

  <!-- Aquí hi hauria la relació de temes d'interès de la persona....-->
    <foaf:topic_interest> tema d'interès per la persona </foaf:topic_interest>
    <foaf:topic_interest> tema d'interès per la persona </foaf:topic_interest>
    <foaf:topic_interest> tema d'interès per la persona </foaf:topic_interest>

  <!-- Aquí faig la descripció de la feina actual de la persona....-->
    <foaf:currentProject>
      <foaf:name>elNom de la Feina que realitza</foaf:name>
      <foaf:Agent>
        <foaf:name> el nom de l'empresa Actual</foaf:name>
        <foaf:workplaceHomepage rdf:resource="www.LaPaginaDeLaFeina.es"/>
        <foaf:phone rdf:resource="t933333333"/>
        <foaf:Person>
          <foaf:title> gerent </foaf:title>
          <foaf:name> el nom de la persona de contacte </foaf:name>
        </foaf:Person>
      </foaf:Agent>
    </foaf:currentProject>

  <!-- Aquí faig la descripció de la feina anterior de la persona....-->
    <foaf:pastProject>
      <foaf:name>elNom d'una Feina que va realitzar</foaf:name>
      <foaf:Agent>
        <foaf:name> el nom de l'empresa anterior</foaf:name>
        <foaf:workplaceHomepage rdf:resource="www.LaPaginaDeLaFeina.es"/>
        <foaf:phone rdf:resource=":9344444444"/>
        <foaf:Person>
          <foaf:title> gerent </foaf:title>
          <foaf:name> el nom de la persona de contacte </foaf:name>
        </foaf:Person>
      </foaf:Agent>
    </foaf:pastProject>

```

```

<!-- Aquí faig la relació de coneguts de la persona....-->
  <foaf:knows>
    <foaf:Person>
      <foaf:givenname>Guillermo</foaf:givenname>
      <foaf:surname>Escribà Martínez </foaf:surname>
      <foaf:mbox_sha1sum>e9b49bc8beeea0651aba37aa675b4d</foaf:mbox_sha1sum>
    </foaf:Person>
    <foaf:Person>
      <foaf:givenname>Antonio</foaf:givenname>
      <foaf:surname>Martínez Martínez </foaf:surname>
      <foaf:mbox_sha1sum>d7860cdfdc80e970ee3b1cc</foaf:mbox_sha1sum>
    </foaf:Person>
    <foaf:Person>
      <foaf:givenname>Carlos</foaf:givenname>
      <foaf:surname>Barber Senao</foaf:surname>
      <foaf:mbox_sha1sum>928e693cbc975a57194</foaf:mbox_sha1sum>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
</rdf:RDF>

```

L'objectiu del disseny d'aquest segon tipus de document que correspon als grups és el de tenir lligam amb la gestió pròpia d'un centre d'estudis on un grup d'alumnes reflexa un conjunt de persones que realitzen els mateixos estudis dins d'un any acadèmic. Per aquesta raó l'identificador de node d'un grup és la contracció de la del tipus d'estudis, el curs (majoritàriament primer i segon ) i l'any acadèmic. Posteriorment es dona valor al nom del grup i apareix un conjunt d'elements member, que són elements persona on es dona el valor de l'atribut nodeId que referencia als elements de l'altre tipus de documents, el dels alumnes.

Un document FOAF d'un grup d'alumnes tindria el següent esquema:

```

<?xml version="1.0" encoding="ISO-8859-1" standalone="yes" ?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">

<foaf:Group rdf:nodeID="DAI12004-05">
  <foaf:name> 2 Desenvolupament d'Aplicacions Informàtiques curs 2004-05</foaf:name>
  <foaf:member>
    <foaf:Person rdf:nodeID="000001"/>
  </foaf:member>
  <foaf:member>
    <foaf:Person rdf:nodeID="000002"/>
  </foaf:member>
  <foaf:member>
    <foaf:Person rdf:nodeID="000003"/>
  </foaf:member>
  <foaf:member>
    <foaf:Person rdf:nodeID="000004"/>
  </foaf:member>
</foaf:Group>

</rdf:RDF>

```

Les col·leccions s'organitzaran de la següent manera:

Una col·lecció que representarà el cursos, és a dir, tots els documents de cada grup d'alumnes. i un altre col·lecció que representarà als alumnes.



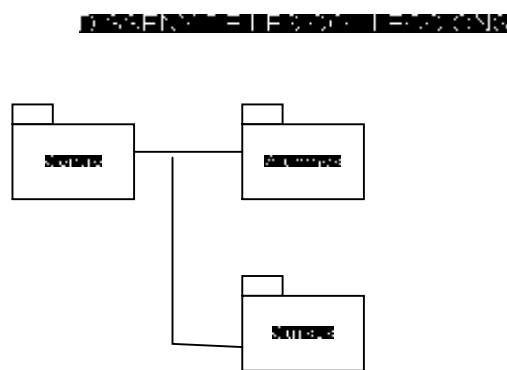
Per a millores en l'aplicació el fet de matricular a un alumne implica tenir el seu document FOAF dins la col·lecció alumnes i inclòs com a membre d'un document de tipus grup ubicat dins de la col·lecció cursos.

Quant comencem un curs caldrà crear documents de tipus grups d'alumnes.

Aquests grups tindran un atribut nodeID dins de l'element foaf:Group que pendrà valor a partir d'una concatenació obtinguda de la següent manera:

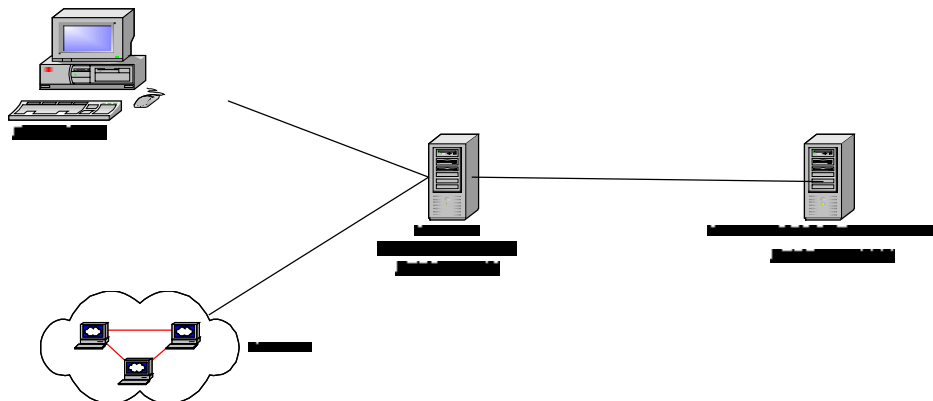
- Contracció del nom del tipus d'estudis p.e. a Desenvolupament d'Aplicacions Informàtiques li correspondrà DAI
- Curs 1 o 2
- Any acadèmic: 2004-05 ó 2005-06....

Posteriorment per a la matriculació d'alumnes caldrà incloure els seus nodeID dins d'un element foaf:Person dins de foaf:member. Si s'actualitzen les dades d'un alumne, aquestes quedaran actualitzades per a tots els grups ens el que ha estat membre.



## 11.4 Implementació.

El procés d'implementació es basarà en implementar les consultes XQuery descrites i posteriorment incloure-les en una aplicació integrada i amb GUI implementada en PHP emprant codi HTML



Per a visualitzar correctament l'aplicació caldrà configurar el browser per a que la codificació de caràcters correspongui amb el tipus Unicode UTF-8.

A continuació podem donar un cop d'ull a la relació del nom dels arxius implementats amb el nom d'aquests que s'ha emprat en la fase d'anàlisi :

#### Cas d'Ús nº 1: AUTENTICACIÓ

Nom arxiu implementat	Nom fase d'anàlisi
index.php	PantallaAutenticacio
Control_acces.php	
pral.html	Menu
titol.html	
menu.html	

Nota: el procés d'autenticació no es fa via SOAP ja que ho fa internament el propi arxiu control\_acces.php

#### Cas d'Ús nº 2: CONSULTA FITXA D'UN ALUMNE

Nom arxiu implementat	Nom fase d'anàlisi
intro_consulta_alumne.html	Pantalla_intro_alumne
mostra_alumnes_per_fitxa.php	Pantalla_seleccio_alumne
selalumne.php	Pantalla_fitxa_alumne
eXist_soap.php	clientSoap

#### Cas d'Ús nº 3: CONSULTA D'ALUMNES PER TEMES D'INTERÈS

Nom arxiu implementat	Nom fase d'anàlisi
consulta_temes.html	Pantalla_intro_temes
mostra_alumnes_tema.php	Pantalla_seleccio_alumne
selalumne.php	Pantalla_fitxa_alumne
eXist_soap.php	clientSoap

#### Cas d'Ús nº 4: CONSULTA DELS AMICS D'UN ALUMNE

Nom arxiu implementat	Nom fase d'anàlisi
consulta_amics_alumne.html	Pantalla_cerca_amics
mostra_alumnes2.php	Pantalla_seleccio_alumnes
mostra_coneguts_alumne.php	Pantalla_alumne_i_amics
eXist_soap.php	clientSoap

#### Cas d'Ús nº 5: OBTENIR LLISTATS DE MATRICULACIÓ

Nom arxiu implementat	Nom fase d'anàlisi
intro_consulta_curs.html	Pantalla_llistat_curs
consulta_llistat.php	Pantalla_seleccio_alumnes_curs
selalumne.php	Pantalla_fitxa_alumne
eXist_soap.php	clientSoap

#### Cas d'Ús nº 6: CONSULTA D'ALUMNES PER EMPRESA

Nom arxiu implementat	Nom fase d'anàlisi
intro_consulta_empresa.html	Pantalla_alumnes_per_empresa
cerca_alumnes_empresa.php	Pantalla_seleccio_alumnes
selalumne.php	Pantalla_fitxa_alumne
eXist_soap.php	clientSoap

#### Cas d'Ús nº 7: OBTENIR ESTADÍSTIQUES DE MATRICULACIÓ

Nom arxiu implementat	Nom fase d'anàlisi
consulta_matriculacio.phpl	Pantalla_estadistica
eXist_soap.php	clientSoap

#### Cas d'Ús nº 8: CONSULTA D'EMPRESSES PER ALUMNE

Nom arxiu implementat	Nom fase d'anàlisi
consulta_empreses_alumne.html	Pantalla_cerca_empreses_alumne
mostra_alumnes_obtenir_empreses.php	Pantalla_seleccio_historial_alumne
mostra_historial_alumne.php	Pantalla_historial_alumne
eXist_soap.php	clientSoap

En la present aplicació totes les imatges s'ubiquen dins d'un directori anomenat *imatges* dins de l'anomenat *html* del servidor Apache.

L'adreça doncs serà [http://www.elmeuservidor.com/imatges/nom\\_imatge](http://www.elmeuservidor.com/imatges/nom_imatge)

Degut a que l'aplicació s'ha fet funcionar en local, en l'arxiu `eXist_soap.php` s'ha modificat creant una variable `wsdl` que comunica amb `http://localhost:8888/exist/services/Query?wsdl` pren la seva informació i la utilitza per a crear una instància d'un `SoapClient`, és a dir, construeix la petició SOAP.

El WSDL és l'equivalent d'un resum en XML – describint els serveis Web, on s'ubiquen, i com es poden invocar. Per aquesta raó s'ha tingut que carregar l'arxiu `Query.xml` dins del subdirectori `$/CATALINA_HOME/exist/services`. En ell es descriuen en XML les definicions dels tipus, missatges, tipus de port, operacions i serveis WSDL

Dins del directori adjunt ***jperezbrAplicacioTFC*** podem veure la següent estructura de directoris

- **html:** on s'ubiquen tots els arxius html i php per a que funcioni l'aplicació aquests arxius s'hauran de copiar al directori `/var/www/html` si aquest és el directori principal del servidor Apache
- **imatges:** on s'ubiquen totes les imatges que es poden emprar per a visualitzar l'exemplificació de les consultes en que es demana la ubicació del nom de l'arxiu de la fotografia de l'alumne. Aquest directori es copiarà sencer i s'en crearà un altres dins de `/var/www/html` amb el mateix nom., és a dir `/var/www/html/imatges`.
- **services:** aquest directori conté l'arxiu `Query.xml` de manera que aquest arxiu s'ubicarà dins de `$/CATALINA_HOME/webapps/exist/services` que és on es farà la petició soap en cas de fer-ho funcionar en local. En cas contrari s'haurà de modificar el valor de la variable és a dir la línia on es veu:

`$this->wsdl = "http://localhost:8888/exist/services/Query?.wsdl"`

per una d'externa

- **centre:** on es troben els directoris `cursos` i `alumnes` on s'ubiquen els documents `rdf:foaf` de l'aplicació
- **backup:** backup de la base de dades que m'ha servit per desenvolupar i fer les proves de l'aplicació

## 11.5 Valoració econòmica.

---

Degut a que és una aplicació que té com a objectiu demostrar mitjançant un exemple pràctic de bona part del que he descrit en el present Treball Final de Carrera i que s'ha utilitzat software gratuït caldria valorar el temps emprat en el desenvolupament de l'aplicació i de la posta a punt dels requeriments previs de configuració, la qual cosa no te sentit ja que surt de l'objectiu del treball que és aprendre. Però si que es podria fer per a implementar una aplicació, evidentment més completa, que permetés enregistrar als alumnes de tots els centres d'estudis ubicats en un territori en el que l'ús de les TIC tindria que tenir una forta implantació. Aquesta aplicació podria utilitzar-se pels centres d'estudis i pel propi Departament d'Educació per a la gestió de la matriculació i pel seguiment de la tasca formativa en el centre de treball (pràctiques). No obstant considero que la valoració d'aquest últim punt sobrepassa l'abast del present treball.

## Capítol 12 Conclusions

Primerament haig de dir que podem considerar que al voltant de XML hi ha un conjunt de noves tecnologies que estan en continua expansió i evolució pel que suposa l'intercanvi d'informació en sistemes distribuïts com és el cas de l'ús d'Internet

A partir de XML han nascut, o s'han independitzat, altres llenguatges com és el cas de XSL, XPath, XQuery i sembla ser que això continuarà sent així ja que aquesta evolució és força intensa

Podem veure l'empremta de l'ús de XML en la configuració dels molts serveis així com en la definició de llenguatges per l'intercanvi d'informació. Entre ells podem veure l'ús de XML en arxius de configuració de Linux i cada vegada més en la definició de llenguatges emprats en aplicacions web ja que és senzill d'interpretar per un ordinador i no gaire complicat per a ser entès pels éssers humans

Els sistemes gestors de bases de dades relacionals més coneguts van adquirint noves possibilitats de tractament de dades XML però sempre realitzant aquest tractament sota el concepte relacional i emmagatzemant les dades com a tals, com és el cas d'Oracle. Per altra banda comencen a aparèixer, encara que en una fase primigènia, els anomenats XML:DB que tracten directament documents amb format XML

La principal característica de les XML:DB és la capacitat d'obtenir resultats de consultes amb format XML. Són bases de dades i com a tals suporten transaccions, accés multiusuari, llenguatges de consulta, indexació i d'altres característiques dissenyades especialment per emmagatzemar documents XML

Dins d'aquests sistemes de bases de dades, els XML:DB document-cèntrics he avaluat tant *Xindice* com *eXist* i puc dir que ara per ara *eXist* facilita molt més les tasques d'administració i ús sobretot per la versatilitat d'utilització de diferents interfícies, la utilització de XQuery, i per poder ser acollit tant el propi Jetty que porta incorporat com per Tomcat o d'altres.

La utilització de XML arriba al punt de que els ordinadors es puguin comunicar entre ells sol·licitant serveis que desconeixen i de manera transparent a l'usuari. Això són els anomenats serveis UDDI i s'ha vist que moltes empreses estan interessades en poder intercanviar informació, sobre tot aquella informació que tingui un interès corporatiu. Els missatges SOAP utilitzen XML per a les preguntes com per les respostes, així com també s'utilitza en l'estructura de definició dels serveis que ofereix WSDL.

Per últim cal dir que he exemplificat tot això en el desenvolupament d'una aplicació que caracter corporatiu que funcioni a internet emprant browser com a client i un servidor web Apache, on s'ubica l'aplicació, de manera que aquest últim realitza consultes a un servidor Tomcat, on s'ubicaria *eXist* com a sistema XML:DB. Aquestes consultes es realitzarien mitjançant Soap i WSDL i demostrar la comprensió final del que he exposat en el present treball realitzant consultes XQuery hostatjades en codi PHP per a generar codi HTML i poder ser visualitzat i permetent la interacció amb l'usuari

## Bibliografia

### Llibres

---

“**XML A través de ejemplos**”. *Abraham Gutiérrez i Raúl Martínez*.  
Editorial Ra-Ma 1ª edició 2001

“**XML Manual de referencia**”. *Heather Williamson*.  
Editorial McGraw-Hill 1ª edició 2001

### Documents

---

“**Extensible Markup Language (XML) 1.0** . *Tim Bray, Jean Paoli, C.M. Sperberg-McQueen*.  
versió <http://www.w3.org/TR/REC-xml> Actualització 04/02/2004

“**Resource Description Framework (RDF) primer** ”. *Ora Lassila, Ralph R. Swick*.  
versió <http://www.w3.org/TR/rdf-primer/> Actualització 10/02/2004

“**RDF Vocabulary Description Language 1.0: RDF Schema**”. *Dan Brickley, R.V. Guha*.  
versió <http://www.w3.org/TR/rdf-schema/> Actualització 10/02/2004

“**FOAF Vocabulary Specification**”. *Dan Brickley, Libby Miller* .  
versió <http://xmlns.com/foaf/0.1/> Actualització 02/09/2004 revisió 1.66

### Altres adreces web

---

<http://www.linuxparatodos.net/linux/como-java-linux.php>

<http://www.osmosislatina.com/tomcat/instalacion.htm>

<http://www.sidar.org/recur/desdi/traduc/index.php>

<http://exist-db.org/index.html>

<http://xml.apache.org/xindice/>

<http://f14web.com.ar/inkel/2003/01/27/foaf.html#foafnaut>

<http://www.ldodds.com/foaf/foaf-a-matic.es.html>

<http://www-106.ibm.com/developerworks/xml/library/x-foaf.html>

<http://www.foaf-project.org/>

<http://rdfweb.org/topic/FoafExamples>

[http://www.zvon.org/xxl/XPathTutorial/General\\_spa/examples.html](http://www.zvon.org/xxl/XPathTutorial/General_spa/examples.html)

<http://www.w3.org/TR/xquery-use-cases/>